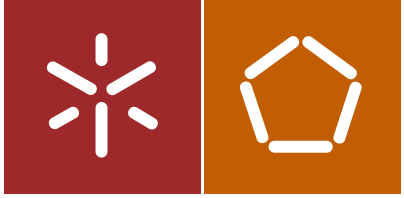
Universidade do Minho
Escola de Engenharia

Daniel Martins da Silva

Analytics-as-a-Service no Contexto de Plataformas de Big Data para Smart Cities

Analytics-as-a-Service no Contexto de Plataformas de Big Data para Smart Cities
Daniel Martins da Silva
UMinho | 2016

outubro de 2016



Universidade do Minho
Escola de Engenharia

Daniel Martins da Silva

Analytics-as-a-Service no Contexto de Plataformas de Big Data para Smart Cities

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao Grau de
Mestre em Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação do
Professora Doutora Maribel Yasmina Santos

outubro de 2016

DECLARAÇÃO

Nome: Daniel Martins da Silva

Endereço eletrónico: a66921@alunos.uminho.pt Telefone: 914369452

Número do Bilhete de Identidade: 12791719

Título dissertação*: Analytics-as-a-Service* no Contexto de Plataformas de *Big Data* para *Smart Cities*

Orientadora: Professora Doutora Maribel Yasmina Santos

Ano de conclusão: 2016

Designação do Mestrado: Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação

Universidade do Minho, ___/___/_____

Assinatura:

The best way to predict your future is to create it.

(Abraham Lincoln)

AGRADECIMENTOS

Uma dissertação na sua génese é um processo solitário a que qualquer investigador está destinado, no entanto este reúne o contributo de várias pessoas, aproveito pois para agradecer a todas aquelas que de uma forma, ou de outra, contribuíram para a minha aprendizagem tanto a nível pessoal, como profissional, ao longo desta jornada.

Aos meus pais, por todo o apoio que sempre demonstraram e confiança depositada na minha pessoa, estando sempre do meu lado em todas as decisões e ocasiões da minha vida.

Aos meus familiares, na generalidade por se preocuparem em relação à minha evolução, em especial ao meu tio Rui Sá que me ajudou ao longo de todo o processo de revisão deste documento. À minha prima Suse pelo seu precioso contributo e disponibilidade na revisão final.

À colega Sara Veloso, pelo tempo despendido, pela troca de comentários e experiências nesta jornada.

Ao Fábio, pelo seu contributo tanto neste documento, como na força e exemplo de uma jornada deste tipo.

Ao Carlos, pelo seu contributo e disponibilidade que sempre demonstrou.

Ao professor João Varajão, pela disponibilidade e ajuda que me concedeu no UML.

Um especial agradecimento ao colega de curso Bruno Martinho, pelo apoio, pela troca de experiências e conhecimento partilhado.

Reservo pois, como não poderia deixar de ser um agradecimento singularmente especial para a minha orientadora na pessoa da professora Maribel Yasmina Santos, e que pelas limitações da linguagem escrita e da minha capacidade de a descrever cinjo-me a breves palavras de apreço à sua pessoa, pelo rigor, profissionalismo, simpatia, dedicação, pelo que não só correspondeu como excedeu as minhas expetativas em toda esta dissertação.

Por último, mas não menos importante, um agradecimento especial para o meu amigo Hélder Martins que me apoiou e instruiu em algumas temáticas, sem nunca olhar a horários ou a fins de semana.

A todos estas pessoas e a outras que não referi, como colegas de curso, amigos e familiares, professores, ao grupo TMG pela abertura e apoio, e às pessoas que tomaram parte e que sabem que lhes tenho um agradecimento especial, pois revelaram-se importantes no percurso universitário que agora encerro.

Obrigado!

RESUMO

O fluxo migratório que se assiste nas últimas décadas, nomeadamente para os centros urbanos, tem como consequência problemas sérios de sustentabilidade, tanto ao nível de recursos naturais, como ao nível da qualidade de vida dos seus habitantes. Sabe-se que 75% da população da União Europeia vive em cidades, e calcula-se que até 2020 este número suba para os 80%, o que preocupa não só as autoridades centrais, como locais. Por outro lado, verifica-se uma mudança de mentalidade, em que o cidadão demonstra interesse em colaborar com as autoridades, para e desta forma se iniciar uma governação participativa. Os desafios que são colocados a este tipo de gestão tornam a mesma impossível por meios tradicionais. No entanto, a proliferação das novas tecnologias já demonstrou a sua mais-valia nas mais variadas áreas, desde que estas sejam bem harmonizadas. Este projeto de dissertação tem como objetivo providenciar a arquitetura bem como a plataforma necessária para colmatar esta necessidade analítica através de um serviço disponibilizado no paradigma *as-a-Service.* Foi realizado um enquadramento conceptual, sobre a literatura relevante, dando ênfase aos conceitos de *Big Data*, plataformas, exemplos e arquiteturas de *Big Data*, tanto ao nível de tratamento como de análise de vastos volumes de dados, dando uma visão do estado de arte em relação à temática em estudo. No que diz respeito ao enquadramento tecnológico foi apresentada a arquitetura BASIS (Arquitetura de *Big Data* para *Smart Cities*), que constitui o ponto de partida para esta dissertação, dando continuidade ao trabalho já realizado e seguindo a necessidade identificada pelo autor de aprofundar a camada analítica da arquitetura. Neste sentido foram realizadas análises às plataformas *Pentaho*, *BIRT*, *SpagoBI* e *Jaspersoft*, de uma forma detalhada, por forma a serem percecionadas as principais características de cada uma delas com vista a identificar a que melhor se enquadra e que responda aos requisitos da arquitetura BASIS. Após a realização do estado da arte foi estabelecida uma arquitetura tecnológica que permitiu a execução de pequenos testes à plataforma *SpagoBI* onde se constatou que esta era capaz de suprir parte das necessidades analíticas. Seguidamente é feita a proposta de detalhe analítico para a arquitetura BASIS, dando particular atenção à camada conceptual e tecnológica da proposta, colmatando uma das principais lacunas encontradas na literatura, a falta de detalhe tecnológico. Por fim, a proposta de arquitetura foi validada através da integração de funcionalidades *SpagoBI* no protótipo "SusCity" dando exemplos de utilização do serviço.

Palavras-Chave: *Analytics-as-a-Service, Big Data* e *Smart Cities*.

ABSTRACT

The migration we are witnessing in recent decades, particularly for urban areas, results in serious problems of sustainability, both in terms of natural resources, as in the quality of life of its habitants. It is known that 75% of the EU population lives in cities, and it is estimated that by 2020 this number will rise to 80%, which concerns not only the central authorities but also, the local ones. On the other hand, it´s possible to see a change of mentality, in which citizens show interest in collaborating with the authorities, and thus to start a participatory governance. The challenges are posed to this type of administration, it impossible to make by traditional ways. However, the proliferation of new technologies has proved its added value in various areas, since they are well harmonized. This dissertation project aims to provide the architecture as well as the necessary platform to fill these analytical needs through a service available in the paradigm as-a-Service. In this work, a literature review on the relevant literature was developed, giving emphasis to the concepts of Big Data, platforms, examples and architectures of Big Data, both in terms of treatment and analysis of vast amounts of data, giving important insights on the state-of-the-art in relation to the topic under study. With regard to the technological environment, it is showed the BASIS architecture designed to support Big Data in a Smart Cities context, which is a starting point for this thesis, continuing the work already done, and following the need identified by the author to go into detail on the analytical layer of the architecture. After this, a detailed analysis was carried out to several platforms like Pentaho, BIRT, SpagoBI and Jaspersoft, in order to see their main features and identify the one that best fits and that responds to the BASIS architecture requirements. After the analysis of the state-of-the-art, it was established a technological architecture that allowed the execution of brief tests to the SpagoBI platform where it was found that it was able to make available many of the needed analytical tasks. The proposed analytical detail specified in the BASIS architecture is, paying particular attention to the conceptual and technological layer, thus fulfilling a major gap in the literature, the lack of technological detail. Finally, the proposed architecture has been validated through the integration of the SpagoBI features in the SusCity prototype, using some user examples.

Keywords: Analytics-as-a-Service, Big Data and Smart Cities.

ÍNDICE

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

API – Application programming interface;

BASIS – Arquitetura de *Big Data* para *Smart Cities;*

BDW – Big Data Warehouse;

BI – Business Intelligence;

CRUD – Create, read, update e delete;

DAaaS – Data Analytics-as-a-Service;

DSRM – Design Science Research Methodology;

DSS – Decision Support Systems;

ETL – Extraction, Transformation and Loading;

HDFS – Hadoop Distributed File System;

HQL – Hive Query Language;

HTTP – Hypertext Transfer Protocol;

IEEE – Institute of Electrical and Electronics Engineers;

IoT – Internet of Things;

J2EE – Java 2 Platform, Enterprise Edition;

JDBC – Java Database Connectivity;

JSON – JavaScript Object Notation;

JSP – Java Server Pages;

NoSQL – Not Only Query Language;

ODBC – Open Database Connectivity;

OGC – Open Geospatial Consortium;

OLAP – Online Analytical Processing;

KPI – Key Performance Indicator;

RAM – Random Access Memory;

RDF – Resource Description Framework;

RDMS – Relational Database Management System;

REST – Representational state transfer;

SOA – Service Oriented Architecture;

SQL – Structured Query Language;

XML – Extensible Markup Language.

1.INTRODUÇÃO

Neste capítulo é apresentado o enquadramento e motivação para a realização desta dissertação, assim como a abordagem metodológica, a finalidade, os objetivos e a estruturação do documento.

1.1. Enquadramento e Motivação

Nos dias de hoje cerca de 75% dos Europeus vivem em áreas urbanas, sendo de esperar que este valor aumente para 80% até 2020. O contínuo aumento das populações que vivem nessas áreas faz com que exista a necessidade de aumentar a resiliência das cidades, nomeadamente nos seus mais variados recursos, sendo este um dos principais desafios que se colocam a estes aglomerados populacionais. Manter o desenvolvimento urbano, económico e, por outro lado, assegurar a sustentabilidade dos recursos tais como a água, a energia, entre outros, requer um melhor planeamento que deve ser decidido a nível local (Khan, Anjum, & Kiani, 2013).

O surgimento do conceito *IoT (Internet of Things)* e a sua materialização que vemos a surgir nos dias de hoje promoverá fortemente a proliferação de ambientes em que as coisas apesar de comunicarem entre elas poderão também comunicar com os indivíduos, suportando-os no seu dia-a-dia. A comunidade técnica *IEEE* (*Institute of Electrical and Electronics Engineers*) define o *IoT* como sendo um sistema em rede autoconfigurável e adaptável, constituído por sensores e outros objetos inteligentes com o objetivo de se interligarem. Incluímos nestas coisas não só os objetos do nosso dia-a-dia, mas também os utilizados na indústria, de modo a torná-los inteligentes, programáveis e mais capazes de interagir com os seres humanos. Cidades e regiões equipadas com esta tecnologia permite que as mesmas se tornem autoconscientes do seu ambiente podendo adaptar-se e configurar-se em tempo real ou quase real. Desta forma estarão melhor preparadas, respondendo às adversidades que podem surgir (Boulos, 2015).

Os grandes volumes de dados gerados por todo este ecossistema precisam de ser geridos e analisados para que possam gerar valor. Para isso é necessário utilizar uma abordagem integrada. Existem já várias ferramentas que atuam em diversos domínios. No entanto, não surgem numa perspetiva integrada que permita lidar com a sustentabilidade e crescimento socioeconómico da cidade. As *Smart Cities* podem beneficiar através de serviços interoperáveis numa solução *Cloud*. No entanto, a utilização desta informação necessita de

tecnologias apropriadas para recolher, armazenar, analisar e visualizar grandes quantidades de dados a partir do ambiente da *Smart City* (Khan, Anjum, Soomro, & Tahir, 2015).

Vários são os projetos de *Smart Cities* espalhados por todo o mundo (Shelton, Zook, & Wiig, 2015). No entanto, a componente analítica dos mesmos é um dos pontos ainda pouco aprofundado. Uma das maiores *Smart Cities* da Europa está instalada em Santander, tendo a sua arquitetura e plataforma de suporte sido desenhadas com certos objetivos, como lidar com dados em tempo real e ser flexível para poder lidar com diferentes escalas e tipos de dados. Os autores identificam que é necessário ter em atenção diversos aspetos e apesar dos desenvolvimentos apresentados, é necessário melhorar a semântica dos dados (Bin, Longo, Cirillo, Bauer, & Kovacs, 2015). Na cidade de Bandung na Indonésia foi implementado outro projeto de *Smart City*, no qual os autores apontam como investigação futura, a possibilidade de refinar a componente analítica, de modo a que previsões ou outras técnicas analíticas associadas aos *DSS (Decision Support Systems)* possam ser incorporadas (Suakanto, Supangkat, Suhardi, & Saragih, 2013).

Em (Costa, 2015) podemos encontrar uma arquitetura que se intitula de BASIS, no trabalho desenvolvido este identifica várias arquiteturas existentes identificando o potencial e limitações de cada uma delas. Deparou-se, pois, com a necessidade de criar uma arquitetura de raiz, que visa eliminar a ambiguidade tecnológica encontrada nas anteriores arquiteturas, ao mesmo tempo que cumpre os princípios estabelecidos, destacando a relevância das tecnologias de armazenamento, processamento e análise de *Big Data,* para tal adotou uma abordagem de abstração por camada dividida em camada conceptual, tecnológica, infraestrutural e intervenientes na cocriação de serviços. Concluindo que alguns dos pontos da arquitetura necessitavam de ser validados nomeadamente o papel do *Big Data Analytics* disponibilizado no paradigma *"as-a-Service"* presente na arquitetura.

Dada a relevância da componente analítica no contexto de uma *Smart City* e dando continuidade a proposta futura apontada em (Costa, 2015), na qual são recolhidos, armazenados e processados dados de várias fontes, este trabalho investiga a concretização de mecanismos analíticos oferecidos no paradigma *as-a-Service*, quer do ponto de vista das infraestruturas existentes, quer da proposta de um serviço para uma plataforma de *Big Data* no contexto de *Smart Cities.*

1.2. Abordagem Metodológica

Ao nível da abordagem metodológica, esta secção apresenta, por um lado, a metodologia de investigação adotada para a concretização deste trabalho e, por outro, o processo de revisão bibliográfica necessário para verificar o estado da arte na área.

1.2.1. Metodologia de Investigação

A metodológica de investigação que será seguida para o desenvolvimento desta dissertação será o *Design Science Research* (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), pois esta permite construir um artefacto que vá de encontro aos objetivos e enquadramento desta dissertação. Este modelo mostra-se bastante flexível, pois permite que seja iniciado em vários pontos de partida, garantindo disciplina, rigor e a transparência que é necessária na condução deste tipo de projetos.

Na Figura 1 verifica-se o ciclo de vida da metodologia proposta no *Design Science Research,* podendo verificar-se que esta se encontra dividida em 6 fases:

- *Identify problem & motivate* – Nesta primeira fase é definido o âmbito do problema a ser resolvido demonstrando a sua importância e percebendo quais os recursos necessários para a sua realização;
- *Define objectives of a solution* – Aqui são definidos quais os objetivos a serem alcançados, que podem ser quantitativos ou qualitativos. Verifica-se o estado da arte em relação ao problema proposto;
- *Design & Development* – Passa pela criação do artefacto que engloba todos os modelos e métodos necessários para a solução do problema, podendo conter propostas de novos modelos que melhor se ajustem ou que resolvam o problema proposto de uma forma mais adequada;
- *Demonstration* – É feita a prova de conceito utilizado o artefacto criado anteriormente e aplicando-o ao problema demonstrando a sua viabilidade.
- *Evaluation* – É observado e avaliado se o artefacto cumpre os objetivos para a solução proposta através da comparação do que foi planeado com o que foi efetivamente comprovado;
- *Communication* – Os resultados são demonstrados através de publicações ou apresentações públicas demonstrando a sua importância.

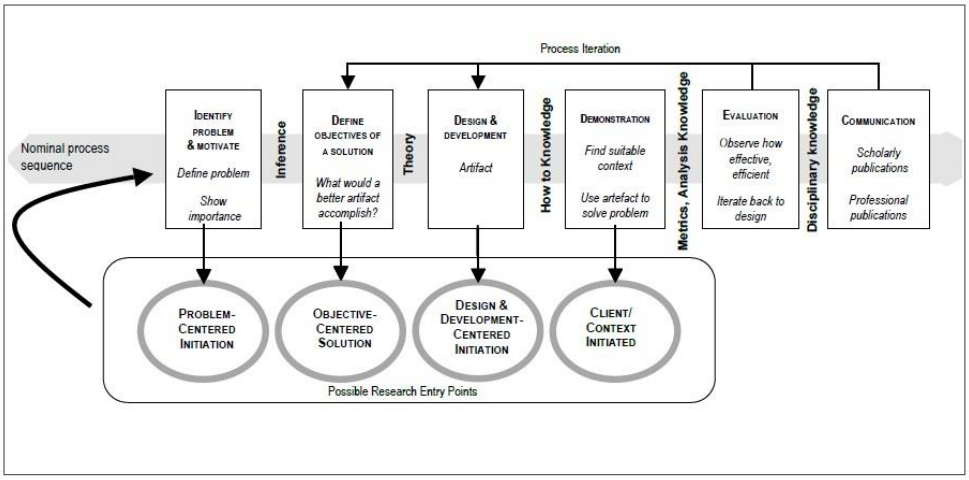


Figura 1 – *DSRM (Design Science Research Methodology) Process Model*. Retirado de (Peffers et al., 2007).

1.2.2. Processo de estruturação do enquadramento conceptual

O desenvolvimento da revisão de literatura efetuada no âmbito deste trabalho centrou-se na pesquisa de informação objetiva sobre o tema e foi realizada sobre fontes consideradas de referência e conceituadas. Para tal, em primeiro lugar, foi necessário identificar as principais palavras-chave em relação ao tema aqui abordado.

As palavras-chave resultam da questão de investigação que se pretende responder neste trabalho: Como concretizar um *Analytics-as-a-Service* no contexto de plataformas de *Big Data* para *Smart Cities*?

Como ilustra a Figura 2, foi dado em primeiro lugar especial atenção a todo o material aconselhado e disponibilizado pela orientadora sendo que esse não sofreu qualquer filtro. Foram destacadas três palavras-chave para a pesquisa, *Analytics-as-a-Service, Big Data* e *Smart Cities*, sendo que estas foram pesquisadas separadamente. No entanto, o universo era bastante alargado e era necessário introduzir especificidade para que estes fossem passiveis de análise, tendo sido feitas pesquisas que tratassem o tema como um todo utilizando várias conjugações possíveis para as três palavras-chave. Foram encontradas algumas dezenas de artigos pelo que foi necessário reduzir o número de artigos a analisar, tendo sido considerados conteúdos com data igual ou superior ao ano de 2010, salvo algumas exceções especificadas posteriormente. Esta pesquisa foi feita em motores de busca conceituados de referenciação/indexação nomeadamente *Scopus, IEEE Xplore, Web Of Science*. No que diz respeito a artigos em revista, foram considerados os classificados no *SCImago Journal & Country Rank* com quartil Q1 e Q2 no último ano de análise se a citação de referência for posterior a 2014 (último ano de análise

do *SCImago*). No caso de revistas Q3 as mesmas também deveriam estar referenciadas no *Web Of Science*, e ter mais de 5 citações (em média ter pelo menos 1 citação por ano).

Em relação aos *conference papers,* foram considerados os referentes ao horizonte temporal em estudo e que cujas conferências fossem realizadas anualmente até à data de hoje.

O critério do número de citações foi muito pouco usado no caso de artigos de conferências, devido a serem publicações muito recentes e, como tal, com poucas citações.

Foi ainda dada especial atenção às referências com texto completo. Quanto às restantes foi fator exclusivo quando apenas se tinha acesso ao *abstract* e este se encontrava pouco claro, não referindo informação concreta sobre o assunto em estudo e demonstrando claramente os resultados obtidos do mesmo.

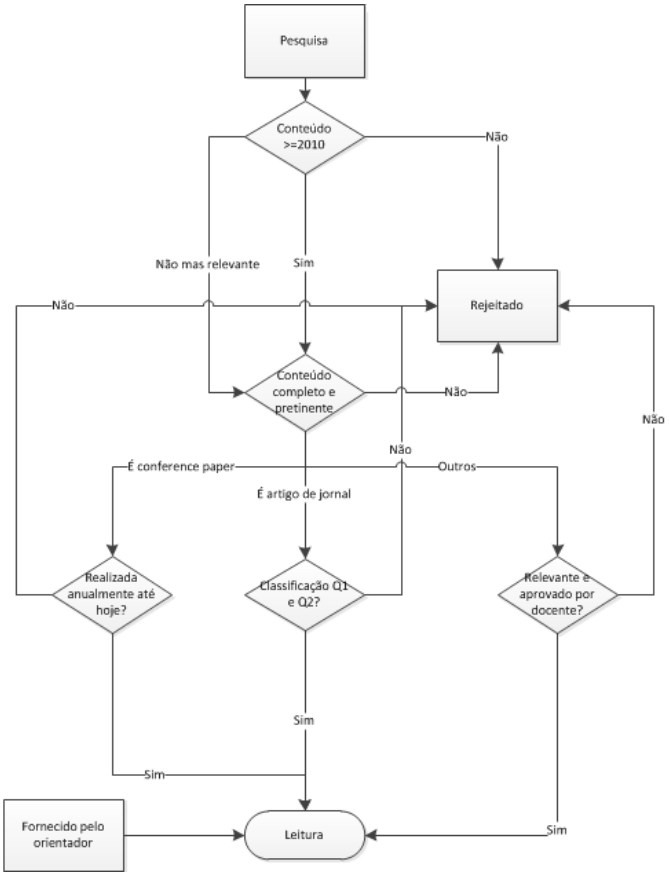


Figura 2 - Processo de seleção de literatura.

1.3. Finalidade e Objetivos

O trabalho a ser desenvolvido nesta dissertação está focado no tema *Analytics-as-a-Service* verificando e explorando as práticas correntes na área, assim como na proposta de um serviço analítico genérico que possa ser oferecido no paradigma *as-a-Service*.

A finalidade desta dissertação passa pela proposta e implementação de um sistema analítico que possa ser utilizado no paradigma *as-a-Service* no contexto de uma *Smart City,* tentando com este responder à questão: Como concretizar um *Analytics-as-a-Service* no contexto de plataformas de *Big Data* para *Smart Cities*? Para ser possível a concretização desta finalidade, são definidos como objetivos deste trabalho:

- Realizar a revisão de literatura sobre os principais tópicos tratados neste trabalho, tais como, *Big Data*, *Data Analytics-as-a-Service* e *Data Analytics* para *Smart Cities*;
- Identificar as principais limitações dos serviços analíticos oferecidos no contexto de *Smart Cities*;
- Propor uma arquitetura de um serviço analítico para *Smart Cities*;
- Implementar a arquitetura proposta na forma de protótipo;
- Analisar e avaliar os resultados obtidos.

1.4. Estrutura do Documento

Este documento está estruturado em 5 capítulos, de seguida brevemente caracterizados. O capítulo 1 consiste na introdução, onde se identifica o problema de investigação, a abordagem metodológica e os objetivos associados a esta dissertação. O capítulo 2 enquadra conceptualmente os termos *Big Data, Analytics-as-a-Service e Smart City*, terminando com a análise *Data Analytics* no contexto *de Smart Cities* já existentes. No capítulo 3, enquadramento tecnológico, está apresentada a arquitetura BASIS assim como a comparação de plataformas analíticas para *Big Data*, finalizando como a experimentação da plataforma *SpagoBI*. No capítulo 4 *Data Analytics-as-a-Service* para *smart cities*, é apresentada e descrita a proposta de detalhe analítico para a arquitetura BASIS, são descritas as principais funcionalidades do serviço proposto, culminando com a sua parcial implementação, teste e avaliação, na forma de protótipo. Por fim, no capítulo 5 são tecidas as conclusões finais tiradas do trabalho apresentado neste documento.

2. ENQUADRAMENTO CONCEPTUAL

Este capítulo pretende dar o enquadramento necessário aos temas de relevo sobre a área em estudo, para que estes sejam entendíveis e passíveis de serem analisados, com vista a perceber o que já foi feito na área em estudo. Este foi assente numa revisão de literatura maioritariamente proveniente da comunidade científica e técnica envolvida nesta temática. Para tal serão abordados temas como *Big Data,* dando uma noção dos conceitos gerais, algumas arquiteturas existentes e *Big Data Analytics*. De seguida introduz-se o conceito de *Data Analytics-as-a-Service,* os seus componentes e arquiteturas. Por fim é abordado o *Data Analytics* para *Smart Cities* fazendo referência ao conceito de *Smart Cities* e seus contextos analíticos.

2.1. *Big Data*

Com o aumento substancial da adoção das tecnologias de informação por parte de todos os agentes económicos, os dados provenientes de diversos setores de atividade têm sido incrementados a um ritmo notoriamente exponencial. Estes dados tendem a ser produzidos de forma cada vez menos estruturada e mais do tipo semiestruturados e não estruturados. Nesta realidade estamos perante aquilo que se intitulou de *Big Data,* onde as tradicionais ferramentas de processamento de dados, como as bases de dados relacionais e *SQL (Structured Query Language)* não conseguem dar resposta a esta nova realidade (Zikopoulos & Eaton, 2011).

2.1.1. Conceitos Gerais

Segundo (Ishwarappa & Anuradha, 2015) *Big Data* é algo extremamente complexo e vasto em que as tradicionais formas e ferramentas de se lidar com dados simplesmente não funcionam. Podemos dizer que *Big Data* é uma coleção complexa de dados, que inclui grandes volumes de dados, que são gerados das mais variadas formas e meios, daí a sua complexidade, volume e heterogeneidade, e que podem ser criados por máquinas, por humanos ou pela natureza. Podem ser provenientes de redes sociais, de sensores, de *logs* e de muitas outras fontes. Constatamos, assim, que as fontes de onde provêm estes dados são extensivamente heterogéneas.

Em 2001, (Laney, 2001) refere as 3 primeiras características de *Big Data,* mais conhecidas como sendo os 3Vs. Neste trabalho, o autor faz referência ao que se faz sentir no quadrante do *e-commerce*, por exemplo, a complexidade do que estava a acontecer, à oportunidade que estava a surgir, sendo necessário criar uma nova visão dos dados. Para tal

apresenta as 3 dimensões, como podemos ver na Figura 3, o volume, a velocidade e a variedade, como forma de encarar esta nova realidade que estende as abordagens tradicionais proporcionando um maior retorno.

Os autores (Zikopoulos & Eaton, 2011) também destacam estas 3 características demonstrando o salto que existiu e que motivou o novo paradigma, *Big Data*, onde observamos que a variedade advém da introdução de dados não estruturados aos já existentes dados estruturados, e a velocidade em que os dados passam da sua generalidade em *batch* para dados em *stream*. Por consequência das duas características anteriores passamos dos Tera bytes para os Zeta bytes endereçando assim o volume (Figura 3).

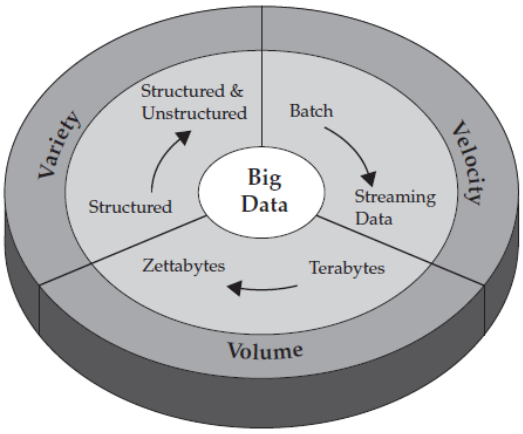


Figura 3 - Caraterização da IBM do modelo 3Vs. Retirado de (Zikopoulos & Eaton, 2011).

Ainda no que diz respeito à velocidade referimo-nos ao facto de os dados serem gerados cada vez mais rapidamente, o que faz com que este aumento não possa ser processado das formas convencionais como, com as bases de dados relacionais. Velocidade refere-se, também ao processo de geração de novos dados, como sendo algo muito rápido, e como estes se movimentam. Por exemplo, as mensagens nas redes sociais podem tornar-se virais em segundos. Em 1999, o *Wal-Mart's Data Warehouse* guardava 1TB de dados, mas em 2012 o valor estava nos 2.5PB de dados. A cada minuto, de cada dia, são feitos *uploads* de centenas de horas de vídeos no *Youtube*, são enviados mais de 200 milhões de *e-mails*, entre muitos outros dados (Zikopoulos & Eaton, 2011).

Em relação à variedade, nem sempre os dados são estruturados e não é fácil colocar grandes quantidades de dados, sendo estes semiestruturados e não estruturados, em bases de dados relacionais. Isto quer dizer que para se analisarem estes dados é necessário conhecer a complexidade dos mesmos e, ter a noção que 90% dos dados gerados não são estruturados (Zikopoulos & Eaton, 2011).

Na mesma linha de pensamento, (Demchenko, Grosso, Laat, & Membrey, 2013) descrevem as características anteriores e acrescentam mais duas passando dos 3Vs para 5Vs, como podemos observar na Figura 4. Os autores começam por descrever o volume como sendo o principal desafio das infraestruturas convencionais, sendo o aspeto que é destacado quando se fala em *Big Data*. Muitas empresas já dispõem de um vasto conjunto de dados em forma de *logs* mas não têm a capacidade de processar esses dados. O benefício de ter essa capacidade de processar enormes quantidades de dados é a principal atração do *Big Data Analytics*.

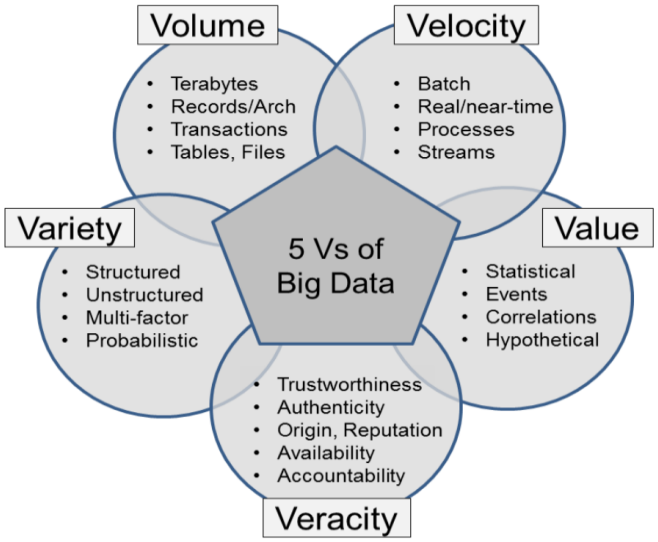


Figura 4 - Modelo dos 5Vs. Retirado de (Demchenko et al., 2013).

O incremento dos 2Vs é descrito pelos autores, como uma forma de tornar mais claro o conceito. Um deles é a veracidade, uma vez que quando lidamos com um grande volume de dados, que são gerados a uma grande velocidade, é impossível garantir que estes estão 100% corretos. A qualidade e a fiabilidade dos dados que são recolhidos têm impacto na qualidade e na veracidade das análises que daí resultarem.

Em relação ao valor, este apresenta-se como sendo o mais importante de todos os Vs. O potencial do *Big Data* só se manifesta quando este gera valor e se torna útil. A implementação de uma infraestrutura capaz de lidar com *Big Data* é bastante custosa e é necessário que esse investimento tenha o devido retorno (Demchenko et al., 2013).

Na mesma perspetiva em (Khan, Uddin, & Gupta, 2014), os autores incrementam duas características às cinco já apresentadas, ficando o conceito assim descrito com 7Vs, e acrescentam ao que já foi dito alguns aspetos importantes. Estes autores descrevem os 7Vs na seguinte perspetiva (Khan et al., 2014):

Volume - Quando se fala em volume em *Big Data* referimo-nos ao tamanho dos dados criados por todas as fontes, incluindo textos, áudio, vídeo, redes sociais, estudos de pesquisa,

medicina, imagens do espaço, relatórios policiais, previsões de tempo e desastres naturais entre outros. Contudo, todo este volume de dados está desorganizado e, na sua maioria não poderão ser tratados das formas tradicionais ou através de simples *queries*. Para uma parte destes dados não estruturados, simplesmente não existe forma de os tornar estruturados e coloca-los em sistemas *RDMS (Relational Database Management System)* como *SQL Server, MySQL* ou Oracle. Não nos podemos esquecer que estamos a lidar com Peta bytes de dados não estruturados, em que uma abordagem *SQL* simplesmente não funciona.

Velocidade - Não importa apenas ter em atenção a velocidade com que os dados são criados, mas também a velocidade com que estes devem ser processados e disponibilizados para uma atempada tomada de decisão. Simplesmente, por vezes, é necessária informação *on the fly* para que esta seja útil ou para uma tomada de decisão atempada e oportuna.

Variedade - Os dados estão disponíveis em vários formatos, como áudio, vídeo, texto, imagens, sendo possível compreender a sua real complexidade e a principal razão pela qual não podemos utilizar bases de dados relacionais. As pessoas acedem à internet em diferentes *browsers* e enviam dados para a *cloud* também de forma diferente. Não podemos ignorar que a maior parte dos dados advêm da interface de humanos, o que causa erros que não podem ser evitados. Esta variedade tem efeitos diretos na integridade dos mesmos, pelo que quanto mais variados forem maiores serão os erros que estes poderão conter.

Veracidade - A veracidade leva-nos a pensar na confiança que temos nos dados. Leva-nos a pensar a quando a sua exploração no significado dos mesmos. Antes de este V ser considerado, a comunidade assumia que todos os dados que eram criados eram claros e precisos, sendo esta visão herdada dos tradicionais métodos de *Data Warehousing*. Neste momento estamos a lidar com dados não estruturados, provenientes dos *posts* do *Facebook*, do *Twitter*, *Linkedin*, entre outros. Serão todos estes dados de confiança? É vital que se usem ferramentas e algoritmos capazes de efetuarem a limpeza necessária em contexto de *Big Data*, já que a confiança que colocamos nos dados depende sempre da sua origem e qual a sua finalidade.

Validade - A validade tende a ser confundida com veracidade, no entanto, não são o mesmo conceito. Como estamos a falar de quantidades de dados onde os relacionamentos, pelo menos em estágios iniciais, são inexistentes ou muito difíceis de encontrar, é muito importante perceber a relação dos dados com o âmbito que se pretende tratar. É necessário então validar os dados se possível antes de serem consumidos.

Volatilidade - Podemos comparar a política de retenção de dados estruturados aplicada a uma qualquer organização, com o que se passa em *Big Data*. O período de retenção dos dados em *Big Data* não é algo que seja facilmente gerido, o espaço e a segurança podem-se tornar demasiado dispendioso e difícil de implementar e de gerir. A volatilidade torna-se significativa por causa do volume, variedade e velocidade dos dados.

Valor – Como retrata a Figura 5, a convergência dos Vs vai de encontro ao objetivo primário de todas estas preocupações que é extrair o máximo de valor possível dos dados. É necessário olhar atentamente para os dados, perceber o seu verdadeiro valor, que deve exceder o seu custo tanto ao nível de retenção quanto ao nível de gestão.

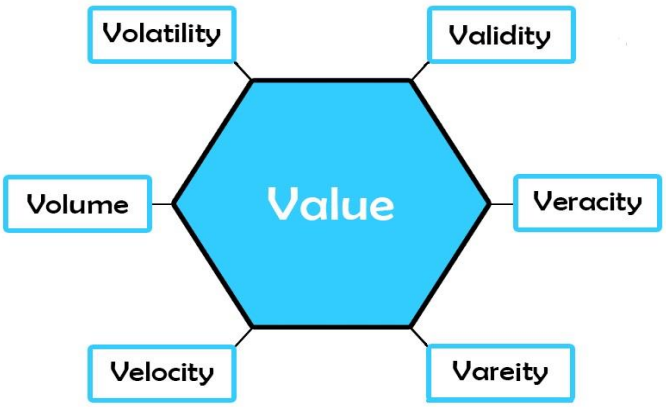


Figura 5 - 7Vs. Adaptado de (Khan et al., 2014).

Tendo já sido caracterizado o conceito de *Big Data,* a próxima subsecção apresenta arquiteturas definidas para a recolha, armazenamento e análise destes dados.

2.1.2. Arquiteturas de *Big Data*

Genericamente, em (Zachman, 1987) define aquilo que é considerado como uma arquitetura de sistemas e que assenta em duas dimensões, uma delas associada às 6 interrogações básicas de comunicação (O quê, Como, Quem, Onde, Quando e Por quê) cruzando com a dimensão de *stakeholders* (Visionário, Dono, Projetista, Construtor, Implementador e Trabalhador), para dar uma visão holística da empresa ou negócio que está a ser modelado.

Em (Sessions, 2007) é definido que uma arquitetura é um esquema que descreve a estrutura de um sistema, representando não só os seus diferentes componentes, mas também explicitando como estes se interligam para atingir um objetivo comum.

Descrito o conceito de arquitetura, de seguida são apresentadas propostas para o contexto de *Big Data.*

Através de (Quintero et al., 2014) a IBM propõe uma arquitetura de *Big Data* que pode ser utilizada em vários domínios, descrevendo a estrutura, os seus componentes, e as relações que enquadram a mesma na família das arquiteturas corporativas. A arquitetura *HPC (High Performance Computing) Big Data* contém quatro camadas, *visual analytics layer, data reduction layer, data model layer* e *data and modeling integration layer*, cujo objetivo é disponibilizar uma base analítica abrangente.

Dentro de cada camada que constitui esta arquitetura, existem componentes chave por onde os dados fluem. A configuração é projetada para grandes volumes de dados sendo que os autores fazem referência a apenas três características das sete já apresentadas, nomeadamente a velocidade, variedade e volume. Na Figura 6 podemos observar as diferenças entre uma arquitetura analítica tradicional e a arquitetura que a IBM propõem para contextos *Big Data*. As principais diferenças são a inclusão de *clusters map reduce, event servers,* e a transição de *selected users* para *masses of users* e de *data warehouse* para *file servers* (Quintero et al., 2014).

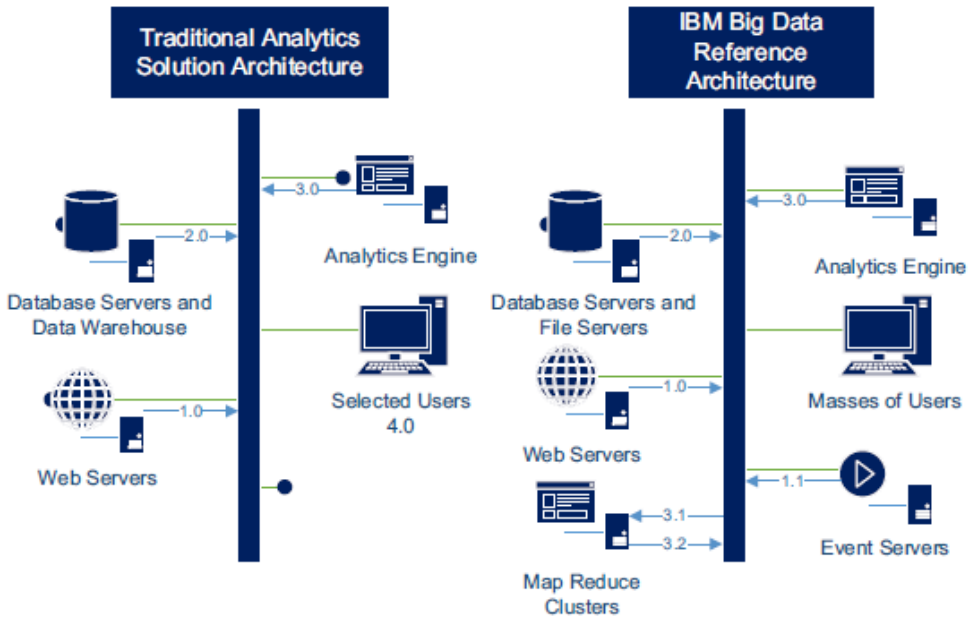


Figura 6 - Arquitetura de referência. Retirado de (Quintero et al., 2014).

A análise em *Big Data* não é composta apenas por uma única ferramenta, mas sim por várias, de entre elas as técnicas de recolha, *Data Mining*, análise preditiva, visualização de dados, *BI (Business Intelligence)*, inteligência artificial, processamento de linguagem natural, e

vários métodos de suporte analítico, como o *Platform Symphony*, *MapReduce*, *in-memory database*, e bases de dados em colunas (Quintero et al., 2014).

Diz (Quintero et al., 2014) que a análise massiva em *Big Data* requer desempenho e escalabilidade, pelo que o problema mais comum que as plataformas tradicionais se deparam esta associado com a sua capacidade de resposta em *datasets* em larga escala. No aspeto de análises de dados, a IBM propõe dois métodos diferentes, guardar e depois analisar *(store-analyse)* ou então analisar e depois guardar *(analyse-store)*.

Na arquitetura proposta pela IBM o *Hadoop* é um elemento chave, sendo utilizado para correr o *Platform Symphony MapReduce* correspondendo ao método *analyse-store*. O *Hadoop* permite a criação de um *cluster* de alto desempenho e escalável onde, e sempre que necessário, se pode acrescentar mais nós sem ser necessário alterar o formato dos dados, nem a forma como os dados são carregados e independentemente da aplicação que esteja a correr em cima. O *Hadoop* tem a característica de absorver qualquer tipo de dados sejam eles estruturados ou não estruturados de diversas fontes.

Como podemos observar na Figura 7, o *Platform Symphony MapReduce* divide a entrada de dados *(input)* em tarefas independentes, iniciadas pela operação de *map* que é executada em paralelo para as várias tarefas. Estas tarefas geram um conjunto de ficheiros intermédios que são posteriormente integrados na operação de *reduce*, encarregue de gerar o ficheiro de saída *(output)*.Esta é então a arquitetura que a IBM propõem para se lidar com *Big Data* (Quintero et al., 2014).

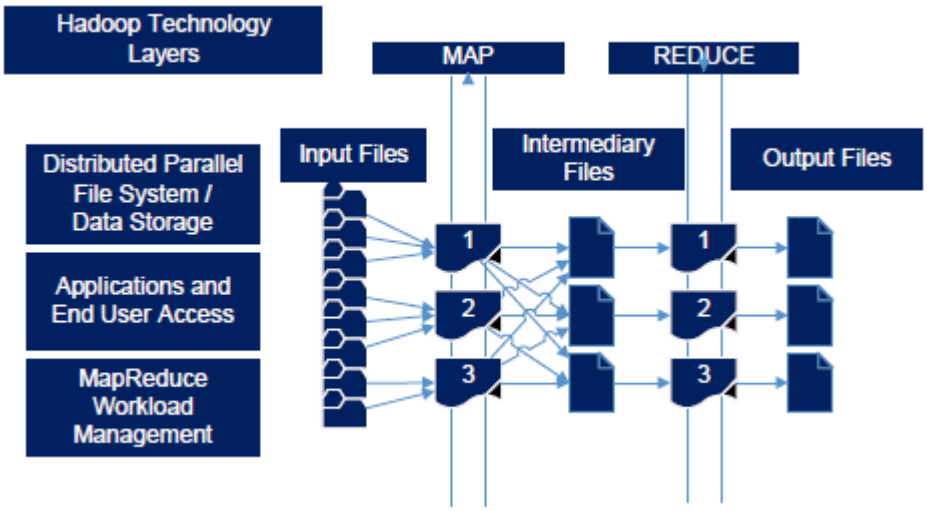


Figura 7 - *Platform Symphony MapReduce*. Retirado de (Quintero et al., 2014).

Uma outra arquitetura, esta proposta pela Oracle em (Oracle, 2015a), refere que uma arquitetura de *Big Data* tem de corresponder aos requisitos de *Big Data,* ou seja, volume,

velocidade, variedade e valor, referindo explicitamente apenas quatros das sete características de *Big Data*. Nesta proposta, diversas capacidades analíticas estão disponíveis para retirar significado de novos dados. O rendimento analítico também tem impacto na transformação, integração, arquiteturas de *storage,* com dados em tempo real ou quase real, exploração e visualização *ad hoc*. No entanto, é comum que após a execução do *MapReduce,* o seu resultado, seja movido para um *Data Warehouse* e/ou para ambientes analíticos dedicados. Na Figura 8 encontra-se representado o modelo conceptual onde se encontram expostos os *discovery labs,* ou ambientes dedicados de análise, que são ambientes fechados, cuja arquitetura é desenvolvida para se adaptar rapidamente consoante as necessidades analíticas.

Neste modelo conceptual a Oracle representa os principais componentes e fluxos. Um de destaque, é a separação e a integração do componente *Discovery Lab* onde são conjugadas formas novas e tradicionais de recolha de dados (Oracle, 2015a).

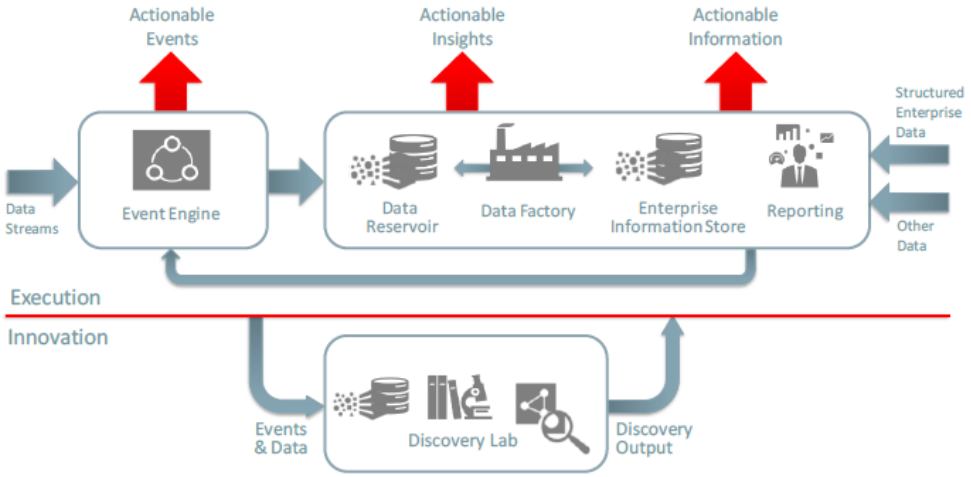


Figura 8 - Modelo conceptual *The Oracle Big Data Platform* para gestão de informação integrada e *Big Data*. Retirado de (Oracle, 2015a).

Neste modelo, os seus componentes podem ser sumariamente descritos como (Oracle, 2015a) define:

- *Event engine*: componente que executa processamento de dados *on the fly*, para identificar eventos que levam à execução de uma ação ou à tomada de decisão, baseada no contexto e no perfil de eventos já existentes e que se encontram guardados em repositórios da empresa;
- *Data reservoir*: aqui encontramos armazenamento escalável e processamento paralelo para dados que não necessitam de uma modelação rigorosa ou formalização. Normalmente contém um *cluster data* ou uma área de estágio numa base de dados relacional;

- *Data factory:* é o orquestrador entre o *data reservoir* e o *enterprise information store*, tendo ainda o papel de garantir o rápido aprovisionamento de dados para o componente *discovery lab* com o objetivo de agilizar a descoberta;
- *Enterprise information store*: repositório de dados relevantes para a tomada de decisão do negócio, tipicamente *Data Warehouse* ou os *Data Marts;*
- *Reporting*: Ferramentas de *BI*, de análise como *dashboards* e relatórios, acesso móvel para comunicação atempada;
- *Discovery lab*: conjunto de componentes composto por repositório de dados, motores de processamento e ferramentas de análise completamente autónomas e separadas das que são usadas diariamente para facilitar a descoberta de conhecimento,

Na Figura 9 é apresentada uma visão holística da arquitetura proposta pela Oracle no contexto de gestão de informação em *Big Data*.

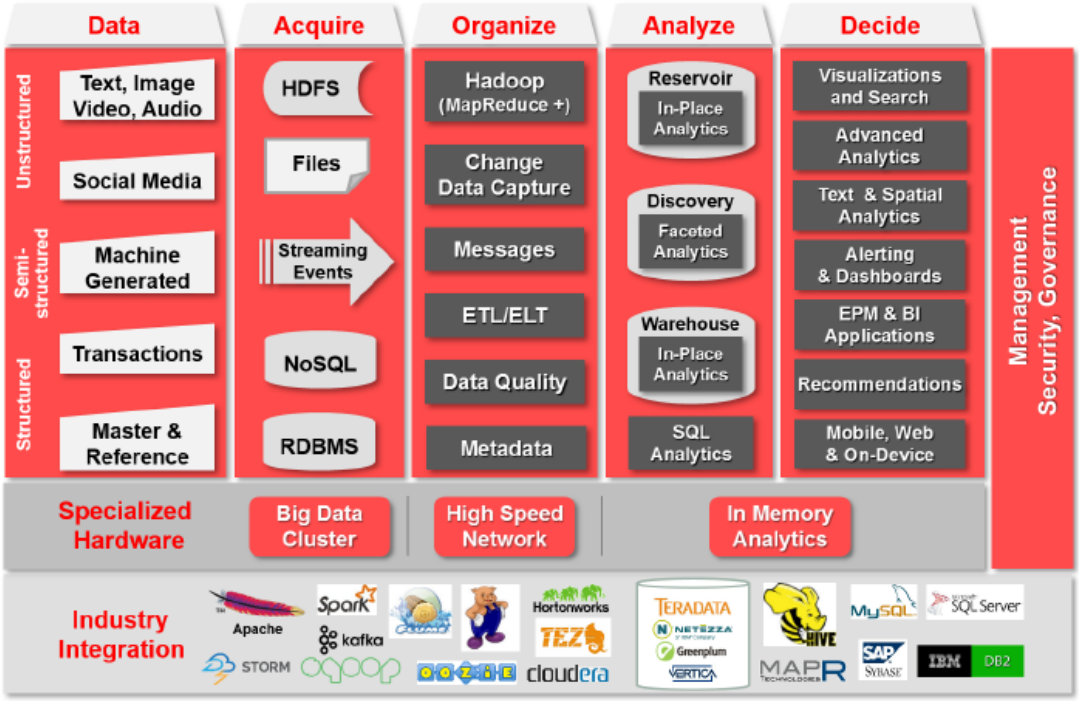


Figura 9 - Oracle's *Unified Information Management Capabilities*. Retirado de (Oracle, 2015a).

Conforme (Oracle, 2015a), e numa visão geral e começando do lado esquerdo da Figura 9, vários tipos de dados podem ser inseridos no sistema através do *acquire,* os quais podem ser escritos diretamente em tempo real em processos de memória ou podem ser escritos em disco em forma de mensagem, ficheiro ou bases de dados. Existem várias opções sobre onde devem

persistir os dados. Podem ser escritos em bases de dados relacionais comuns ou *NoSQL (Not Only Query Language),* ou sistemas distribuídos como *HDFS (Hadoop Distributed File System)*.

É referido ainda em (Oracle, 2015a), no que diz respeito à camada *organize,* esta é caracterizada como sendo uma área ampla e aberta onde se englobam as formas tradicionais e as novas formas de armazenamento e processamento de dados, mais importante, esta área responde aos requisitos dos 4Vs, volume, velocidade, variedade e valor. Além disso permite a verificação da qualidade mantendo os metadados e o rastreio dos mesmos. À saída deste processo, os dados são carregados em *Data Warehouses, data marts, data discovery* ou então no *reservoir.* De notar que o *discovery* requer ligações rápidas com os componentes do *data reservoir, event processing e Data Warehouse,* razão pela qual é necessária uma rede de alta velocidade.

A camada seguinte, *analyse*, é onde os dados são carregados do processamento de *Big Data* que ocorreu anteriormente no *Data Warehouse* para posterior análise. De notar que o processamento analítico pode ser executado tanto no *reservoir* ou no *Data Warehouse* sem nenhum passo adicional o que significa que a análise pode ocorrer sem que seja necessária a passagem para o ambiente analítico. A análise em *SQL* permite combinar *queries* simples ou complexas em cada repositório independentemente, ou combinando resultados em *queries* simples.

Existem várias opções nesta camada que podem melhorar o desempenho no processamento dos dados, como o Oracle Exadata para *Data Warehouse*, processamento em memória, bases de dados em colunas, bases de dados em memória, entre outras.

A camada de *BI (decide)* está equipada com ferramentas que permitem a interatividade, em tempo real. Estas ferramentas têm capacidade de pesquisa e modelação de dados.

Management, security, governance, cobrem todo o espetro tanto ao nível dos dados como da informação ao nível empresarial. É, pois, nesta camada que são e estão definidas as permissões e os controlos adequados, assegurando assim a gestão e coordenação entre as pessoas e as tecnologias.

Com esta arquitetura, o negócio e a sua componente analítica ficam abrangidos com dados prontos para serem consumidos (Oracle, 2015a).

As arquiteturas apresentadas anteriormente, propostas por duas organizações relevantes na área, permitem perceber como estas lidam com o *Big Data*, que componentes são usados, como é que estes interagem entre si, descrevendo as suas características de modo a satisfazer

as necessidades ao nível empresarial. Posto isto, na subsecção seguinte iremos aprofundar o tema *Big Data Analytics* por forma a perceber a sua importância no suporte à tomada de decisão, começando por fazer um breve enquadramento histórico, passando depois por descrever qual o panorama analítico da atualidade e alguns exemplos já implementados.

2.1.3. *Big Data Analytics*

A necessidade de fazer mais com menos nunca acaba (Minneman, 1996), quer no contexto organizacional (organizações públicas ou privadas) quer na sociedade em geral. Já nos anos 80, as tradicionais ferramentas de *BI* estavam focadas em disponibilizar relatórios que descreviam essencialmente atividades passadas. Na última década estas ferramentas evoluíram e estão mais interativas e preocupadas em representar o momento presente. No entanto, nos dias de hoje, os utilizadores procuram olhar para o futuro. A grande quantidade de dados na sua maioria classificados como dados não estruturados coloca novos desafios que se tornam críticos e é necessário preparar a próxima geração de sistemas de *BI* no suporte à decisão (Ng, Arocena, Barbosa, & Carenini, 2013).

O conceito de *analytics* não é novo, existem inúmeras técnicas como a regressão analítica, simulação e aprendizagem automática, sendo que estas já se encontram disponíveis há vários anos. Mesmo a análise de dados não estruturados, como documentos, é algo que já foi amplamente estudado e compreendido. O que é encontrado de novo é a união dos avanços tecnológicos, com *software* e as novas fontes de dados como as redes sociais são geradoras de novas oportunidades de negócio. Com isto foi criada uma nova prática de estudo que tem por nome *data science,* cujo seu objetivo primário é englobar técnicas, ferramentas e tecnologias em processos capazes de retirar valor do *Big Data* (Watson, 2014). É importante reconhecer que o termo *analytics* não é usado consistentemente (Hugh, 2013). Analisando as suas raízes e começando pelos *DSS*, criados em 1970 foram os primeiros sistemas criados no suporte à decisão. Os *DSS* começaram a ser usados academicamente, com o tempo a sua aplicação foi levada para as organizações nomeadamente no *OLAP (online analytical processing)*, sendo que na década de 90 *Howard Dresner* popularizou o termo *BI* sendo este interpretado genericamente como sendo o conjunto de todas as aplicações que suportam a tomada de decisão. A Figura 10 representa a evolução até chegarmos ao termo *analitycs* (Watson, 2014).

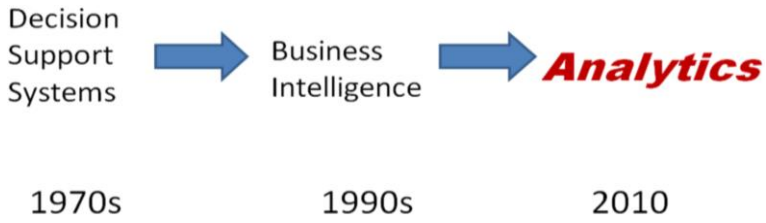


Figura 10 - *DSS* para *BI* para *Analitycs*. Retirado de (Watson, 2014).

Segundo (Tarantola, 2013), no futuro o *Big Data* irá transformar a forma como é analisada uma vasta quantidade de dados. No passado, tentamos compreender o mundo que nos rodeia e as organizações analisando os dados disponíveis, usando análises descritivas que respondiam principalmente a questão "O que aconteceu no passado?". A análise descritiva ajuda as organizações a compreender o passado, sendo que o passado pode ser um minuto atrás ou alguns anos atrás. Por exemplo a análise descritiva ajuda a entender a relação entre clientes e produtos, em que o objetivo é aprender e decidir que abordagem deve de ser adotada no futuro. Por outras palavras, aprender com comportamentos passados de modo a influenciar resultados futuros. Exemplos comuns deste tipo de análise descritiva são relatórios de gestão que fornecem informação de vendas, clientes, operações e finanças, que podem ser analisados e serem encontradas correlações entre as variáveis.

Refere ainda (Tarantola, 2013) que um outo tipo de análises são as análises preditivas, sendo estas uma fonte importante para determinar o que deve de ser feito a seguir, neste tipo de análise os dados podem ser transformados em informações sobre a evolução futura provável de um evento. O *Big data* veio abrir caminho a uma nova era de análises preditivas que estão focadas na questão "O que é que provavelmente irá acontecer no futuro?". Estas oferecem uma estimativa sobre a probabilidade de um resultado futuro. Para fazer isso, uma variedade de técnicas são utilizadas, incluindo aprendizagem automática, *Data Mining* e modelação. As análises preditivas podem por exemplo ajudar a identificar os riscos e oportunidades futuras. Os dados históricos e transacionais são usados para identificar padrões, ao par dos modelos estatísticos e algoritmos que são usados para capturar as relações em vários conjuntos de dados. Com a análise preditiva, é importante ter o máximo de dados possíveis pois possibilita melhores previsões.

As análises prescritivas como sendo o estado final analítico e pode ser chamadas do futuro das análises em que estas pretendem responder a questões como, "E agora?", "Como?", "O que?", "Quando?", onde estas tentam oferecer uma recomendação de decisão baseada no futuro. As análises prescritivas ainda estão num estado muito prematuro, o conceito surge em

2003 e é algo bastante complexo sendo que apenas 3% das empresas usam esta técnica e mesmo assim com vários erros. Em 2013 a consultora Gartner em *Hype Cycle of Emerging Technologies* mencionou que este tipo de análise ainda levará mais cinco a dez anos até atingir a sua plenitude (Tarantola, 2013).

Diz (Rijmenam, 2014) que as análises prescritivas tentam ver o resultado futuro de uma decisão com o objetivo de ajustar essa mesma decisão mesmo antes de essa decisão ser tomada, isto faz com que a decisão seja melhorada. Um dos melhores exemplos deste tipo de análises é o carro sem condutor do Google, este toma decisões baseadas em várias previsões de resultados futuros. Este antecipa os acontecimentos que poderão surgir numa decisão que efetivamente ainda não foi tomada e assim evitar acidentes. Estes tipos de análises poderão ter um impacto significativo na forma como as decisões são tomadas. Estas podem ajudar qualquer organização a se tornar mais eficiente e eficaz.

Segundo o relatório publicado pela *Nucleos Research* (Research, 2014), adicionar capacidades analíticas ao *Big Data* traz enormes vantagens não só competitivas como financeiras na medida em que por cada $1 investido em tecnologias analíticas o seu retorno é de $13.01 (Research, 2014).

O *Big Data Analytics* deve estar no centro da decisão e representa uma oportunidade para as empresas crescerem, serem mais eficientes e preservarem o seu valor. Segundo o estudo publicado pela *Ernst & Young* (Young, 2015), onde foram inquiridos 270 executivos seniores, é referido que 81% dos inquiridos concorda que os dados devem de estar no centro do processo de tomada de decisão, 68% é parte interessada e ativa nos projetos de *Big Data,* 32% admite sentir-se atolado pelos dados, 50% encaram a falta de qualidade dos dados como uma das principais preocupações, 35% reconhece o valor económico do *Big Data* e 10% admite a importância de classificar os dados para que possam ser vendidos ou utilizados numa parceria. O estudo também identifica três desafios a enfrentar, sendo que o primeiro tem que ver com a estrutura organizacional com a criação de uma *framework* de governação com condições para apoiar o processo de decisão orientado para o valor. O segundo desafio está relacionado com a segurança e o cumprimento, o estudo dá conta que 44% dos inquiridos teme que o *Big Data* represente uma ameaça à segurança, 17% apresentam preocupação com a complexidade da regulação e com o risco de incumprimento, e 19% teme que a má utilização ou perda de dados resulte em danos na imagem da empresa. No entanto, uma boa capacidade analítica poderá colmatar tanto a segurança como a necessidade do cumprimento da legislação. O terceiro

grande desafio tem que ver com a retirada de resultados em que os decisores possam atuar (Young, 2015).

O mesmo estudo refere os 10 principais motivos que levam as empresas a implementar o *Big Data Analytics* ordenados por ordem decrescente de importância (Young, 2015):

- Perceber melhor os clientes;
- Melhorar produtos e serviços;
- Melhorar a gestão dos dados já existentes;
- Criar novas fontes de receita;
- Melhorar o modelo de negócio;
- Valorizar monetariamente dados já existentes;
- Gerar eficiências internas de poupança;
- Encontrar e explorar novas fontes de dados;
- Melhorar a gestão de *governance,* do risco e do cumprimento;
- Melhorar a deteção e prevenção de fraude.

Existem já vários exemplos de organizações que tiram partido desta nova realidade e que retiram valor das análises em *Big Data*, um dos exemplos é a previsão da procura e preços da energia.

Como descrito em (IBM, 2012; Jaradat, Jarrah, Bousselham, Jararweh, & Al-Ayyoub, 2015), numa *Smart Grid* onde estão presentes milhões de aparelhos conectados pode-se estimar o consumo de energia. A monitorização de como os consumos de energia está a ser feito pode se tornar uma mais-valia enorme, através da sua análise é possível prever as necessidades de consumo. Esta informação pode ainda ser usada como forma de calcular a quantidade de energia certa para o lugar correto no momento exato. Esta pode ainda ajudar a equilibrar os picos a todo o instante em qualquer lugar. Assim sendo os distribuidores de energia podem melhorar a satisfação dos seus clientes reduzindo o número de interrupções de energia. Se as empresas de energia começarem a analisar e a relacionar o que são as falhas e eventos que acontecem na rede, pode-se começar a entender os padrões que podem indicar problemas, assim como isolar os locais e identificar soluções em tempo real. Quando a *Smart Grid* deteta picos de consumo de energia e se consegue ajustar isso é um ponto fulcral, pois o problema de hoje em dia não é a capacidade energética, mas sim lidar com os picos. Estas redes podem ajudar a minimizar esses picos que podem ocasionar falhas de energia. A análise dos dados de *Big Data* ajudará a otimizar o comércio de energia antecipando a volatilidade dos preços através

da realização em tempo real do mercado com base em milhares de diferentes conjuntos de dados. Prever a oferta e a procura ajudará as organizações a vender a energia de uma forma mais rentável e evitar prejuízos. Ao entender o mercado, podem-se proteger das flutuações dos preços da energia. No final estes são capazes de vender a energia mais barata e aumentar a satisfação dos clientes.

Um outro exemplo encontra-se no setor da saúde onde o *Big Data* ajuda a garantir uma posição financeira saudável a esta organização.

Conforme (Schultz, 2012), *Aurora Health Care* tem 1,2 milhões de clientes, 15 hospitais, 185 clínicas, mais de 80 farmácias comunitárias, e mais de 30000 funcionários, incluindo mais de 6300 enfermeiros e cerca de 1500 médicos o que promove a criação de enormes quantidades de dados. Sendo esta uma organização sem fins lucrativos esta decidiu colocar estes dados com vista a serem usados para uma melhor tomada de decisão e tornar a organização centrada na informação. Em 2012 reuniu em um só *Data Warehouse* dez anos de registos. Combinaram ainda dados do âmbito nacional com os seus próprios dados e resultados e assim criar uma reputação nacional voltada para a qualidade.

Diz (Groenfeldt, 2012) que a empresa utilizou dados clínicos e ferramentas de *Data Mining* para analisar grandes quantidades de dados para poder alcançar melhores resultados. Aurora criou um sistema de *BI* num ecossistema híbrido onde está presente as bases de dados relacionais com o processamento numa plataforma de *Big Data*, aqui processa 18 fluxos de dados em tempo quase real. Isto inclui dados financeiros, das farmácias, de laboratórios e de procedimentos. O objetivo é a utilização de todos os dados de forma altamente segura e eficaz. As tarefas são calculadas utilizando um sistema de processamento paralelo com vários microprocessadores de baixo custo o que dá um poder de computação 20 a 30 vezes superior a um tradicional *Data Warehouse*. Isto permitiu à Aurora olhar de forma diferente para os dados, assim como perceber melhor que pacientes ou grupos de pacientes que têm a mesma doença, como diabetes ou ataques cardíacos. Isto revela novas tendências e resultados que ajudaram os investigadores a encontrar mais facilmente os pacientes certos para testar novos medicamentos. Além disso, o sistema Aurora mantém os registos históricos de cada paciente. Estes dados ficam disponíveis para médicos, enfermeiros e profissionais de saúde em todo o sistema, o que garante que os diagnósticos sejam mais precisos de forma a ser providenciado o melhor tratamento baseado na sua informação pessoal. Este sistema permitiu que a Aurora pudesse tratar mais pacientes em casa, em 2013 enfermeiros equipados com portáteis visitaram

aproximadamente 2300 pacientes. Estes acediam através de ligações seguras aos dados dos pacientes podendo assim avaliar melhor as situações. Usando todos os dados disponíveis e a análise de dados em tempo quase real podem assim prever e melhorar os tratamentos dos seus pacientes. Usando os diferentes fluxos de dados a Aurora diminuiu o número de internamentos de pacientes em 10% o que se traduz em uma economia total de seis milhões de dólares (Groenfeldt, 2012).

Aurora decidiu também juntar-se e participar no desenvolvimento da Oracle na área da saúde, que se traduz em uma plataforma na *cloud* que permite a cooperação entre institutos de ciência, investigadores e profissionais de saúde. Ao se juntar a esta rede a Aurora promoveu a saúde dos seus pacientes através da participação em testes com novas drogas (Schultz, 2012) (Oracle, 2015b).

Os resultados que a Aurora conseguiu alcançar através da adoção do *Big Data* são visíveis pelo já retratado, no entanto é de ressalvar que para além da poupança que conseguiram com a redução dos reinternamentos conseguiram também ainda poupar 42% nos custos de tratamentos (Schultz, 2012).

Um outro exemplo da utilização do *Big Data* com o objetivo de reduzir o impacto ambiental e aumentar a segurança ao nível rodoviário.

Conforme (Tanko & Burke, 2015), o consumo de combustível pode ser reduzido de diversas formas. Em primeiro lugar, os sensores colocados no motor podem monitorizar e otimizar a entrada de combustível. Quando estes são combinados com o melhor percurso tendo em conta as condições meteorológicas, as condições da estrada, comportamentos de condução, uma grande quantidade de combustível pode ser poupada. Os sensores podem também monitorar a velocidade que o condutor conduz assim como se este está a respeitar as regras de transito. Podem monitorar se o condutor já está a conduzir durante muitas horas ou estados de distração, estes podem alertar o condutor e assim evitar acidentes. A cidade de Brisbane na Austrália desenvolveu uma visão completa em tempo real da rede de transportes públicos, que permite simular estratégias num ambiente virtual em tempo real e estável. Esta plataforma permite que a cidade preveja e reduza o congestionamento, resultando assim que os seus passageiros fiquem mais felizes enquanto as emissões de dióxido de carbono são reduzidas. A cidade usa também os limites de velocidade e algoritmos de gestão de filas para melhorar a segurança nas suas estradas.

Por último um exemplo de implementação de tecnologias de *Big Data* na educação.

Diz (Rijmenam, 2014) que as novas tecnologias permitem às escolas, faculdades e universidades analisar absolutamente tudo o que acontece, tanto com os alunos, como com professores e funcionários, desde o comportamento dos alunos, os resultados dos testes para o desenvolvimento de carreira, para as necessidades educacionais em razão da mudança de sociedades. Uma grande parte dos dados já foram recolhidos e são utilizados para a análise estatística das agências governamentais, tais como o Centro Nacional para Estatísticas da Educação.

A universidade de Purdue localizada em West Lafayette, Indiana, com cerca de 40000 alunos e cerca de 6600 funcionários. Foi fundada em 1869 e foi condecorada em 2012 como sendo a mais inovadora ao nível do programa de retenção de alunos. Esta universidade prepara o futuro adotando *Big Data*, e neste momento já alcançou resultados muito significativos.

Esta desenvolveu um sistema com o nome *Course Signals,* que é um sistema que ajuda a prever problemas académicos e comportamentais, notifica professores e alunos quando estes necessitam de executar alguma ação (Arnold & Campbell, 2011). O sistema garante que cada aluno atinja o seu potencial máximo, enquanto diminui a taxa de abandono e reprovação. A plataforma tem sido muito bem-sucedida, em 2012 ganhou o prémio *Lee Noel and Randi Levitz Retention Excellence* (Noel & Levitz, 2012). Esta plataforma é vista como sendo um bom exemplo de como o *analitycs* pode ser aplicado no ensino superior e que ajuda na melhoria de resultados dos alunos, combinando a modelação preditiva com o *Data Mining*. A plataforma usa diversas fontes de dados tais como o sistema de gestão dos cursos e a informação do estudante. A partir da segunda semana de um semestre é capaz de interpretar a preparação de um estudante, o seu empenho e esforço em um determinado ponto no tempo. Para conseguir isto ela usa características dos estudantes e a sua preparação académica, o seu esforço na execução de tarefas como *quizzes* e outros elementos de avaliação bem como o tempo necessário na sua resolução. O algoritmo prevê um perfil de risco para cada aluno através de um sistema visual de semáforos em que o verde significa uma alta probabilidade de sucesso, amarelo onde existem problemas potenciais e vermelho há risco de falha. O fato desta informação estar disponível a partir da segunda semana dá aos alunos a oportunidade de melhorar os seus resultados, promovendo e aconselhando iniciativas para que o aluno possa melhorar o seu desempenho. A plataforma também fornece *feedback* aos professores, e permite que estes sigam os seus alunos e o seu desempenho. Esta plataforma está a ser usada desde 2007 e os resultados são

evidentes, os alunos têm melhores classificações, bem como a universidade melhorou a sua taxa de retenção (University, 2015).

A parceria da universidade com a EMC nomeadamente no fornecimento do espaço necessário para o funcionamento da plataforma. Todos os alunos têm ainda disponíveis 100GB o que corresponde num total de 4PB, trabalhando ainda em conjunto com o objetivo de desenvolverem novas formas de processar, analisar, transferir e gerir o vasto volume de informação (Rigsby, 2012). A *Purdue* identificou que o *Big Data* é muito importante tanto ao nível de pesquisa como de educação. Atualmente esta está muito a frente de outras instituições de ensino na adaptação e implementação de uma estratégia de *Big Data* (Rijmenam, 2014).

2.2. *Data Analitycs-as-a-Service*

DAaaS (Data Analitytics-as-a-Service) representa uma abordagem que converge numa plataforma analítica baseada na *cloud*[1]. Numa perspetiva funcional esta plataforma cobre uma solução fim-a-fim, desde a aquisição de dados até à sua visualização pelo utilizador final. Arquiteturalmente, e devido à complexidade inerente aos processos analíticos, a implementação de um *DAaaS* apresenta um conjunto de desafios, desde logo a sua definição enquanto plataforma, apesar de se assemelhar a um *PaaS (Platform-as-a-Service)* quanto à sua flexibilidade, por outro lado não será tão fechado como *SaaS (Software-as-a-Service)*. Aspetos internos de uma arquitetura do tipo *PaaS*, como a distribuição do processamento, as características de um serviço analítico, a necessidade de guardar dados, modelar e desenvolver modelos híbridos na *cloud* que usam uma base de *cloud* particular, combinada ao uso estratégico de serviços de *cloud* pública, entre outros, torna o seu *design* um desafio complexo (Atos, 2013).

2.2.1. Conceitos Gerais

É definido em (Atos, 2013) que um *DAaaS* é uma plataforma analítica baseada na *cloud*, onde várias ferramentas são colocadas ao dispor do utilizador, as quais podem ser configuradas pelo mesmo de forma a proporcionar um processo de análise mais eficiente em vastas quantidades de dados que podem ser heterogéneos.

[1] No âmbito deste trabalho a computação na *cloud* é assumida como computação a pedido na qual recursos partilhados são disponibilizados aos utilizadores.

Como podemos visualizar na Figura 11 os clientes alimentam a plataforma e recebem da análise, resultados[2], que podem ser padrões, modelos, tendências ou outra informação relevante sobre os dados analisados e potencialmente úteis. Estes resultados são gerados por ferramentas analíticas que são orquestradas por *workflows* analíticos de dados concretos. Estes fluxos de trabalho são constituídos por uma coleção extensível de serviços que implementam algoritmos analíticos, muitos deles baseados em algoritmos de aprendizagem automática, como *Data Mining*. Os dados fornecidos pelo utilizador são ainda complementados com fontes de dados externas. A plataforma *DAaaS* foi desenhada para ser usada em diferentes casos de uso. Por exemplo, o sistema suporta a integração de diferentes fontes de dados externas. Para que o *DAaaS* seja extensível e fácil de configurar a plataforma engloba uma série de ferramentas que suportam o ciclo de vida de todas as capacidades analíticas (Atos, 2013).

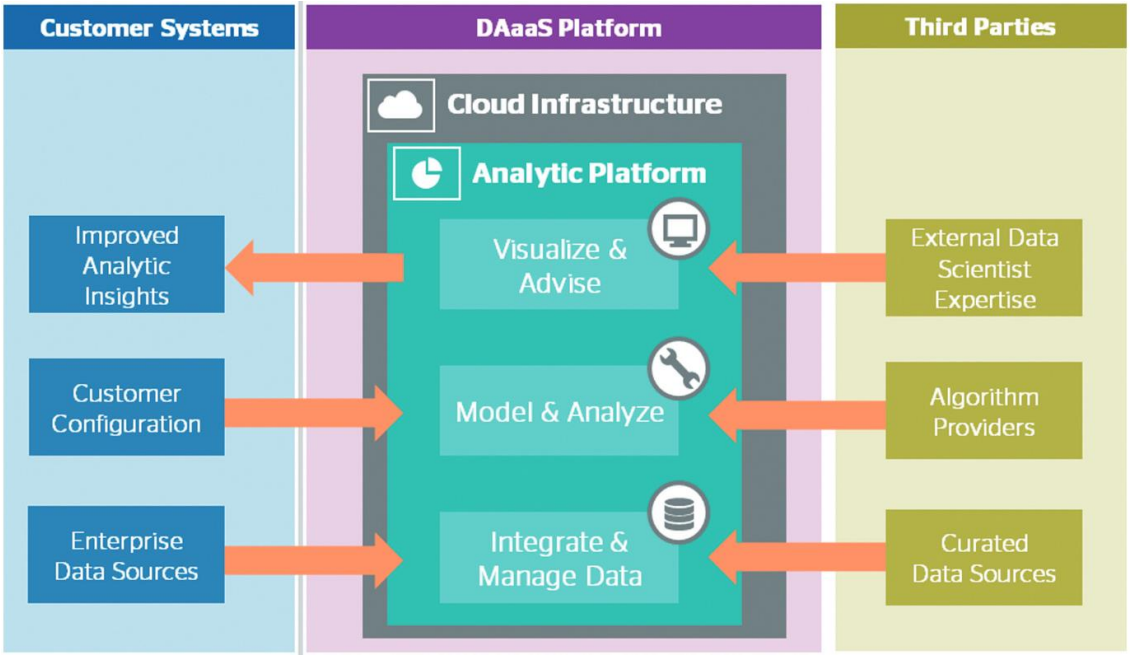


Figura 11 - Conceito *DAaaS*. Retirado de (Atos, 2013).

2.2.2. Componentes do *DAaaS*

Conforme o que está descrito em (Atos, 2013) e a fim de proporcionar as capacidades perspetivadas para uma solução *DAaaS*, uma plataforma que a caracterize necessita de ser implementada considerando os elementos funcionais representados na Figura 12.

[2] O termo utilizado em inglês é habitualmente "*insights*".

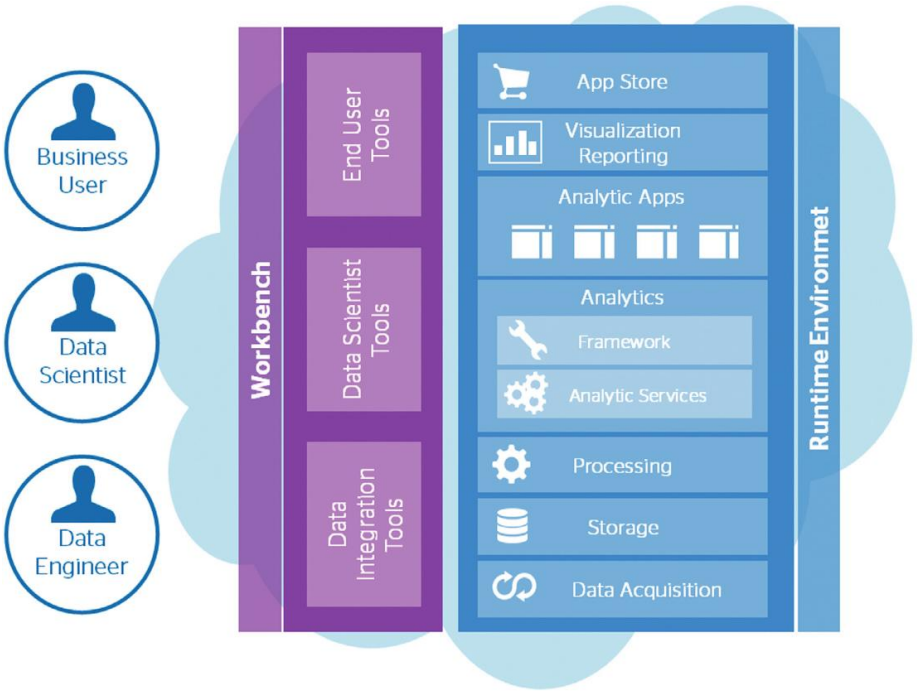


Figura 12 - Elementos funcionais da *DAaaS*. Retirado de (Atos, 2013).

É referido em (Atos, 2013) que nestes elementos funcionais é necessário diferenciar dois grupos de elementos. Elementos relacionados com o tempo de execução da solução, ou seja, a plataforma que processa os dados, identificado como sendo o *Runtime Environment*, e os elementos que controlam a interação com o utilizador, principalmente na configuração do sistema, utilizando para tal um conjunto de ferramentas que estão integradas no *Workbench Environment*.

É referido ainda em (Atos, 2013) que neste último ponto é utilizado um conceito de utilizador bastante amplo, onde não se inclui apenas os utilizadores finais, mas também todos os outros que interagem com a plataforma, como o engenheiro de dados, que faz a modelação e a definição de regras de integração e o cientista de dados que configura o fluxo de dados do serviço analítico.

O *Runtime Environment* é a plataforma de execução da solução *DAaaS*. De seguida é apresentada uma descrição dos vários componentes, como ilustra a Figura 12, seguindo uma abordagem *bottom up* e seguindo a lógica de *input* dos dados para *output* gerado (informação) (Atos, 2013).

- *Data Acquisition:* Providencia um interface de mensagens por *web services* para a aquisição de dados externos. Para fornecer a flexibilidade para lidar com diferentes fontes de dados e protocolos, esta solução necessita de ser

modular com componentes reconhecidos pelo *EIP*[3] *(Enterprise Integration Patterns)*;

- *Storage:* É o repositório de dados, onde a solução deve ser capaz de lidar com Peta bytes de dados, mas também necessita de ser suficientemente flexível para poder albergar diferentes modelos de dados. Existem várias bases de dados *NoSQL* que incorporam em si os recursos necessários;
- *Processing:* Para se processar um vasto volume de dados é necessário utilizar técnicas de processamento distribuído, de forma a serem executados diferentes algoritmos que permitem a sua execução em paralelo. Esta camada de processamento atua como interface entre o armazenamento e os serviços analíticos. Nos dias de hoje, a solução mais popular é o *Hadoop MapReduce*, não só suportado por diversas bases de dados *NoSQL,* mas também por diversas linguagens de programação que se encontram em camadas superiores. Existem, no entanto, outras alternativas emergentes como o *Spark*[4] ou o *Storm*[5] que aparecem com algumas vantagens específicas nomeadamente em cenários em tempo real;
- *Analytics:* Esta é a componente central da plataforma, já que é aqui que os processos analíticos residem. Esta é dividida em dois elementos. Os *Analytics Services*, onde são disponibilizados algoritmos analíticos, que podem ser muito variados e implementados utilizando diferentes bases tecnológicas. No entanto, estes têm um âmbito bem definido para uma determinada análise sobre um determinado conjunto de dados. Muitos destes algoritmos são baseados em técnicas de *Machine Learnning,* usando abordagens supervisionadas e não supervisionadas. Quanto ao *Analytic Framework,* este funciona como integrador, mantendo juntos os diferentes serviços analíticos por forma a produzir o resultado esperado. Aqui um programador especializado poderá usar esta *framework* para implementar uma funcionalidade analítica especifica;

[3] Um EPI é um mecanismo que estabelece *standers* que facilitam a integração de sistemas complexos (Hohpe & Woolf, 2004).

[4] https://spark.apache.org/

[5] https://storm.apache.org/

- *Analytic Apps:* Como já verificado, podemos combinar diferentes serviços analíticos usando as capacidades da *framework* analítica, ao que podemos chamar de leque de aplicações analíticas. Estas aplicações estão orientadas para o negócio e para o utilizador final das mesmas;
- *Visualization/Reporting:* Apesar de uma grande parte das funcionalidades da plataforma *DAaaS* poder ser acedida usando uma interface de *web services,* uma solução completa irá sempre integrar formas de visualização e de *reporting* para simplificar o acesso e interpretação da informação. Existem já um vasto conjunto de aplicações comerciais tais como o *Tableau* e o *QlikView* e outro conjunto de código aberto tais como o *Pentaho* e o *Jasper Reports*;
- *App Store:* Aqui serão providenciados os pacotes de funcionalidades de alto nível ou aplicações para o utilizador final. É também aqui que estarão implementados mecanismos que iram controlar o ciclo de vida das aplicações desde a sua aquisição até ao seu fim de vida. Para que sejam adicionadas ferramentas especificas para responder as especificidades do utilizador final através do *Analytic Apps*. Chamamos então a este conjunto de ferramentas denominado de *Workbench Environment* que incluiu diferentes tipos de ferramentas mediante as permissões definidas;
- *Data Integrator:* Tem a responsabilidade de servir de interface entre o que já existe na organização, nomeadamente outras plataformas de dados, com o *DAaaS*. Aqui é feito o *ETL(extract, transform, load)* onde a informação é modelada de forma a ser integrada no ambiente do *DAaaS*;
- *Data Scientist:* Uma parte substancial do trabalho do *Data Scientist* é modelar e testar a parte analítica, nomeadamente o fluxo de trabalho exposto pelas aplicações analíticas nos *datasets* específicos. As aplicações analíticas ou serviços serão construídos por estes. Estes podem ser funcionários da empresa ou contratados para efetuarem um serviço especializado;
- *Business Users:* Um importante ponto a reter é que existe diferentes tipos de utilizadores que muitas das vezes não têm qualquer conhecimento técnico e que simplesmente esperam um resultado da ferramenta, sejam estes integrados nas ferramentas já existentes ou em novas, em que por vezes a sua simples visualização é o suficiente. No entanto, existem outros

utilizadores que apesar de não terem conhecimentos profundos como um *Data Scientist* têm um papel importante na definição das regras de negócio a serem implementadas.

Depois de identificadas as principais características de um *DAaaS* de seguida passamos a descrever a sua arquitetura.

2.2.3. Arquiteturas para *DAaaS*

Numa arquitetura *DAaaS* e depois de termos abordado as suas principais características e desafios é necessário definir uma arquitetura para uma plataforma deste tipo.

Um *DAaaS* não é um *SaaS* mas sim um *PaaS* especializado (Atos, 2013). Como verificado anteriormente, os passos necessários para a configuração de todas as suas componentes por forma a responder adequadamente a todas as necessidades analíticas a que este se propõe não é tarefa fácil e é necessário ter em atenção os seguintes passos (Atos, 2013):

- Adaptar o processo de *ETL* com as várias fontes de dados;
- Adaptar os modelos do *DAaaS* e os standards dos metadados;
- Modelar e configurar os parâmetros das aplicações analíticas, definindo o *workflow* usando os serviços analíticos;
- Validar os resultados com os utilizadores;
- Integrar os resultados no sistema empresarial;
- Configurar os relatórios e visualizações.

É referido ainda em (Atos, 2013) que a plataforma *DAaaS* é extensível e genérica o que faz levantar algumas questões importantes, nomeadamente no que diz respeito à modelação e armazenamento de dados. Basicamente, a base de armazenamento necessita de ser capaz de guardar diferentes tipos de dados, bem como ser capaz de guardar um vasto volume de dados de forma distribuída, integrar os modelos de dados com a camada de processamento, onde os serviços analíticos sejam capazes de ser processados com o desempenho necessário, fornecer capacidades multiutilizador e satisfazer as necessidades necessárias de segurança. Existem inúmeros motores de bases de dados que podem satisfazer estes requisitos, nomeadamente as bases de dados *NoSQL* onde existem diferentes abordagens, como sendo orientadas a documentos, ao par chave valor, orientadas a colunas, a redes, entre outras. Cada solução tem os seus pontos fracos e fortes, no entanto para uma plataforma *DAaaS* podemos escolher duas opções, selecionar uma base de dados que se adapte a todos os potenciais casos de uso ou

utilizar diferentes tecnologias de bases de dados. A segunda opção adapta-se melhor às potenciais bases de dados distintas e introduz um importante nível de complexidade na arquitetura.

Conforme (Atos, 2013), numa primeira instância, escolher um modelo pode ser difícil. Em todo o caso, a plataforma *DAaaS* foi desenhada para que seja possível a reutilização da camada de modelo de dados com as camadas externas, a camada de aquisição e os interfaces com os serviços analíticos e a camada analítica.

É detalhado em (Atos, 2013) que os serviços analíticos são construídos em blocos, em que a ideia básica é implementar os componentes da arquitetura onde os interfaces com os sistemas externos são definidos claramente para poderem ser implementados. Aqui podemos distinguir diversos tipos de interfaces. Interfaces com a camada de dados, para podermos aceder a informação que estes operam, interfaces que retornam resultados para outro componente ou resultados finais, interfaces que configuram ou modelam a execução dos serviços analíticos, e ainda interfaces por exemplo de monitorização do sistema.

Os serviços analíticos são desenhados para responder a um âmbito específico, o fluxo de trabalho destes é definido nos elementos do *Analytic Framework*. A este fluxo chamamos de *Analytical Apps*. Os componentes dos serviços analíticos e a *framework* foram desenhados para correrem de forma distribuída utilizando as capacidades de soluções como o *Hadoop MapReduce, Spark* ou *Storm*. Tecnicamente os blocos de construção podem ser utilizados de diversas abordagens, alguns deles baseados em estatísticas e técnicas de aprendizagem automática como as redes neuronais, redes de *Bayesian, Support Vector Machines,* análises de regressão entre outras (Atos, 2013).

Um outro aspeto importante e a ter em consideração no *DAaaS* está relacionado com as capacidades de resposta em tempo real. De ressalvar que estamos a falar de respostas na ordem de segundos *(soft real time)*. Normalmente as análises não são feitas com dados em tempo real, no entanto os avanços tecnológicos e as pressões dos negócios de hoje obrigam a respostas rápidas, ao mesmo tempo que o volume de dados cresce exponencialmente. Então a arquitetura deve providenciar mecanismos para uma resposta rápida. Neste caso a plataforma *DAaaS* enfrenta alguns problemas nomeada algumas tecnologias não estão alinhadas com este conceito, a plataforma do *Hadoop* é a mais popular quando estamos a falar de plataformas de *Big Data* no entanto esta não está orientada para este requisito. Por exemplo a *framework* do *MapReduce* é baseada na execução em *batch.* Então é possível termos análises em tempo real

na plataforma *DAaaS*? A resposta é que, por si só não é possível, no entanto pode ser complementada com outros sistemas que disponibilizam a resposta em tempo real com a coordenação e as capacidades do *DAaaS*. A ideia principal está retratada na Figura 13 (Atos, 2013).

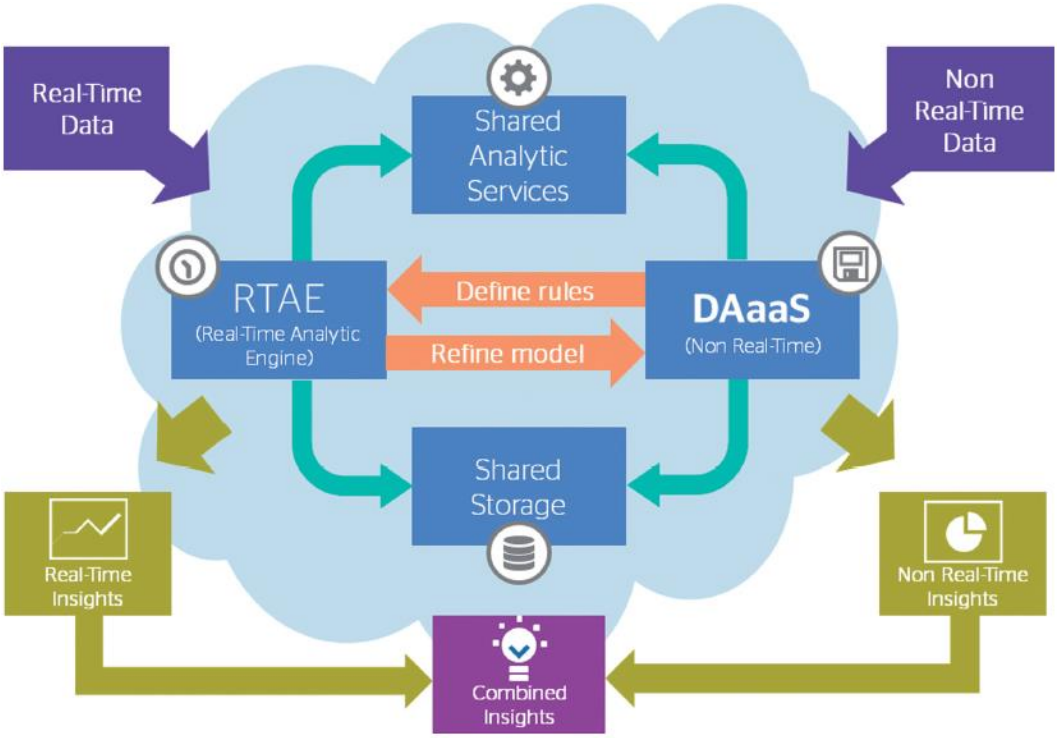


Figura 13 - Análise em tempo real. Retirado de (Atos, 2013).

2.3. *Data Analytics* para *Smart Cities*

Nesta última secção do enquadramento conceptual é dado especial foco na definição do conceito de *Smart Cities* bem como na análise dos contextos analíticos existentes em projetos de *Smart Cities*, dando alguns exemplos que são encontrados na literatura.

2.3.1. Conceitos Gerais

Diz (Gibson, Kozmetsky, & Smilor, 1992) que o conceito de *Smart City* tem atraído o interesse mundial, incluindo governos, empresas, universidades e instituições. Diferentes *stakeholders* tentam entender e explicar o conceito de diferentes pontos de vista. O termo "*Smart City*" surgiu pela primeira vez na década de 90, onde pesquisas davam ênfase à tecnologia, inovação e à globalização em processos de urbanização. Giflinger e Gudrun listam seis características que devem ser consideradas na sua definição, sendo estas a economia, a governança, o ambiente, as pessoas, a mobilidade e a sustentabilidade (Giffinger & Gudrun, 2010). Uma definição de *Smart City* adotada pela *Information and Communication Technology*

tem que ver com construir uma cidade (administração, educação, transporte, etc.) mais inteligente e eficiente (Mitton, Papavassiliou, Puliafito, & Trivedi, 2012).

Segundo (Bronstein, 2009) o conceito de *Smart City* é originário da IBM com a designação *"Smarter Planet"*, sendo que os Estados Unidos da América, pelo presidente Barack Obama, foi dos primeiros países a lançar projetos de *Smart Cities* complementando as noções de *Smart Planet*.

Segundo os autores (Zhang, He, Mao, & Xiao, 2015), no senso comum *Smart City* refere-se à aplicação das mais avançadas tecnologias de informação a cidades modernas, combinada com o *IoT* e a computação na *cloud*. Com esta combinação é possível tomar decisões inteligentes de gestão urbana, fundindo a inteligência artificial com a inteligência humana, criando um ambiente urbano que proporcione uma boa qualidade de vida para os seus habitantes. As tecnologias de uma cidade digital devem possibilitar que organicamente, estas sejam capazes de organizar informação dando-lhe semântica. Para além de agrupar dados relativos à cidade, esta deve permitir que os dados estejam disponíveis a todos os cidadãos, assim como todos os cidadãos, se assim o desejarem, podem contribuir com a sua própria informação em qualquer tipo de terminal e é conveniente que existam diferentes categorias de serviços de informação. As tecnologias de *IoT* permitem alcançar a liberdade de comunicação entre homem e máquina, máquina para máquina e de pessoa para pessoa no contexto de *Smart Cities*.

Diz (Schaffers et al., 2011) que os serviços eletrónicos tornam-se cada vez mais importantes para o desenvolvimento urbano, sendo que as cidades estão cada vez mais a assumir um papel crucial nos motores de inovação em áreas como a saúde, a inclusão social, meio ambiente e empresas. Surge então a questão de como as cidades, suas regiões vizinhas e as áreas rurais devem evoluir para um ecossistema aberto e sustentável de inovação, onde os serviços são orientados para o utilizador.

Segundo (Li, Shan, Shao, Zhou, & Yao, 2013) a expansão das cidades que assistimos nos últimos anos lançam novos desafios que têm de ser respondidos, nomeadamente na melhoria da gestão urbana, as novas tecnologias adaptadas à internet como a computação na *cloud* e a internet das coisas *(IoT).* O conceito de *Smart City* surgiu com a ajuda das novas tecnologias de informação com vista a resolver os problemas de gestão dos centros urbanos. *Smart Cities* têm uma ligação orgânica com dados urbanos e da vida real, a computação e a tomada de decisão através desta enorme quantidade de dados que estão disponíveis numa

plataforma na *cloud* de modo a ser possível a automatização da infraestrutura de controlo urbano.

A definição do conceito de *Smart City* ainda é algo emergente, e continua a não ser completamente claro no que diz respeito as perspetivas dos diferentes *stakeholders* (Nam & Pardo, 2011). É difícil de se formalizar uma definição porque a inteligência de uma cidade pode ser apenas uma simples e única função disponibilizada a um certo grupo de cidadãos ou então algo complexo que envolve todo o processo de administração de uma cidade (Wenge, Zhang, Dave, Chao, & Hao, 2014).

Diz (Yin et al., 2015) que uma definição de *Smart City* deve considerar quatro perspetiva chave: governo, cidadãos, empresas e ambiente. Utilizando uma abordagem sistemática de integração de infraestruturas tecnológicas capazes de providenciar uma governação mais eficiente, cidadãos mais felizes, empresas mais prósperas e um meio ambiente mais sustentável.

2.3.2. Contextos Analíticos em *Smart Cities*

Conforme (Jara, Genoud, & Bocchi, 2014) o porto de Santander, situado a norte da costa espanhola, é a cidade da Europa com maior intensidade de dados. *SmartSantander* conta com mais de 18000 sensores de vários tipos, quando o seu número de habitantes ronda os 180000 (Sanchez et al., 2011). Estes sensores recolhem dados da poluição do ar, ruído e outras condições ambientais. Sensores embutidos no pavimento detetam locais de estacionamento e enviam informação para *displays* espalhados pela cidade de modo a ajudar os condutores a encontrar um local de estacionamento se assim o desejarem. Esta infraestrutura está disponível também para pesquisas, ao mesmo tempo que são utilizadas como um serviço para a cidade.

Refere ainda (Jara et al., 2014) que a *SmartSantander* contempla também uma aplicação móvel denominada de *PaceOfTheCity*, que permite aos cidadãos obterem informações *on-line* de pontos de atração turística, paragens de autocarro, centros comerciais, entre outros locais. O ecossistema definido por esta cidade não tem como único fim servir de laboratório de sensores, redes e arquiteturas da internet do futuro, mas também abrir novas oportunidades, como a interligação de um vasto volume de dados e a computação na *cloud,* e assim ser capaz de fornecer inteligência, ser capaz de compreender comportamentos e até mesmo definir ações de acordo com as informações recolhidas dos diversos objetos inteligentes. Na Figura 14 é evidenciado um exemplo onde é observável as zonas onde estão situadas partes dos sensores que estão espalhados pela cidade, existem, pois, 97 sensores de temperatura e 38 sensores se

tráfego. É visível que a localização dos sensores de tráfego, de ruído e de temperatura é divergente, consequentemente os dados recolhidos não são diretamente relacionáveis, no entanto estes oferecem oportunidades de correlações mais gerais como por exemplo a correlação entre o tráfego e a temperatura média da cidade visto que a temperatura média da cidade é uma variável homogénea (Jara et al., 2014).

Figura 14 - *SmartSantander* (tráfego e temperatura). Retirado de (Jara et al., 2014).

Um outro projeto retratado por (Barga, Ekanayake, & Lu, 2012) intitulado de *Daytona* tem como principal objetivo oferecer um *Data Analytics-as-a-Cloud-Service* acessível a qualquer aplicação cliente através de uma *API (application programming interface) REST (representational state tranfer)*. O foco desta pesquisa é identificar e implementar um serviço que suporte a análise de dados, desenvolver um ambiente *cloud* otimizado para a análise e para a aprendizagem automática e ao mesmo tempo integrar este serviço com ferramentas familiares como a folha de cálculo.

Conforme (Barga et al., 2012) a abordagem consistiu na implementação de dois componentes distintos, um tem que ver com o desenvolvimento de um *add-in* para uma folha de cálculo e o outro corresponde ao serviço *cloud* tanto ao nível de *storage* como ao nível analítico. O *add-in* gere a(s) sessão(ões) entre a *cloud* e o serviço analítico nomeadamente os fluxos de dados entre o cliente e a *cloud*. Este ambiente criado no Excel permite ao analista navegar e recolher amostras dos dados que estão na *cloud*. O analista pode ainda invocar o serviço sem se

preocupar com a troca de dados e com a programação, sendo que o serviço por si só providencia um serviço analítico escalável. Quando um utilizador se conecta ao serviço, este guarda automaticamente os dados do cliente na *cloud* de uma forma transparente. O mesmo pode fazer *upload* dos seus dados juntamente com metadados que servirão de auxílio em pesquisas futuras, sendo que posteriormente outros utilizadores podem fazer *download* destes dados e podem explorá-los como entenderem. O serviço introduz ainda o conceito de *workspace* em três níveis: o primeiro traduz a quem é permitido o acesso aos dados e aos resultados analíticos, em segundo quem tem acesso aos algoritmos analíticos e de aprendizagem automática, por fim aos recursos da *cloud*. Quando um utilizador escolhe um *workspace* específico o serviço devolve os dados mediante os algoritmos e os recursos que este utilizador tem acesso assim como os recursos que este pode consumir. Este serviço contém ainda librarias de *standards* em funções de análise de dados e ferramentas de visualização categorizadas por tipos de análise, incluído a regressão, sumários de dados, *ranking* e outros algoritmos de *Data Mining*. Um utilizador pode ainda escrever fórmulas e *workflows* no Excel (Barga et al., 2012).

Na Figura 15 estão retratados os principais componentes do ambiente de execução do *Daytona* onde os algoritmos implementados tiram partido do Azure e do *MapReduce*.

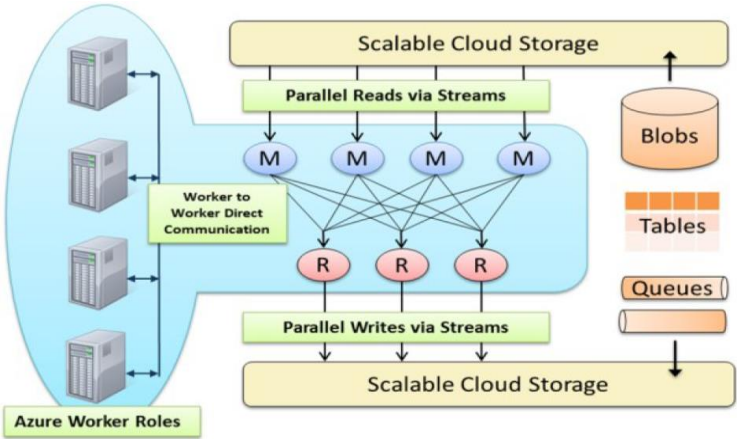


Figura 15 - Principais componentes do ambiente de execução do *Daytona*. Retirado de (Barga et al., 2012).

O serviço analítico é desenhado para ser escalável, dando a possibilidade aos utilizadores de adicionar novas análises e algoritmos de aprendizagem automática ao serviço, registar novos modelos computacionais de simulação e estas funcionalidades são automaticamente disponibilizadas no Excel (Barga et al., 2012).

Uma arquitetura proposta por (Suakanto et al., 2013) com vista ao monitoramento de dados provenientes de vários sensores situados numa *Smart City*. Na arquitetura proposta que se encontra ilustrada na Figura 16, podemos identificar os diversos componentes da mesma, desde os sensores e sua integração na infraestrutura às relações que existem entre os diversos componentes, nomeadamente a integração da rede de sensores na internet e ainda a definição do protocolo de comunicação e a aplicação disponibilizada aos utilizadores.

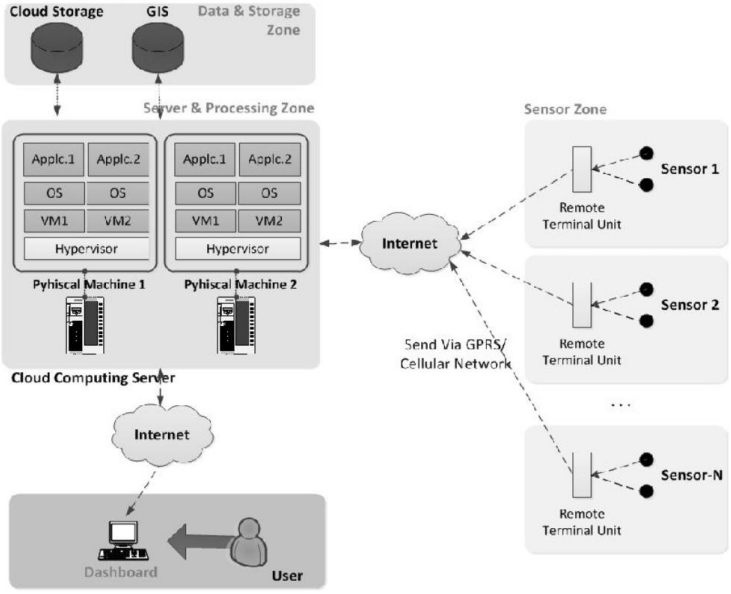


Figura 16 – Arquitetura de rede de sensores. Retirado de (Suakanto et al., 2013).

Esta arquitetura foi testada em contexto real na cidade de Bandung, na Indonésia, com diversos tipos de sensores (temperatura, tráfego, poluição do ar/água) provando assim segundo (Suakanto et al., 2013), que o sistema suporta a implementação de vários tipos de sensores. O protótipo que foi montado pela equipa, refere que foram utilizados apenas um número reduzido de sensores com vista a apenas testar o protótipo.

Como podemos observar na Figura 17, o *dashboard* criado para o protótipo engloba apenas alguns dados sendo que muitos deles, como referem os autores, não advêm de dados reais servindo apenas para demonstrar o conceito. Estes referem ainda como trabalho futuro a refinação da componente analítica para que a arquitetura possa integrar técnicas associadas aos *DSS* (Suakanto et al., 2013).

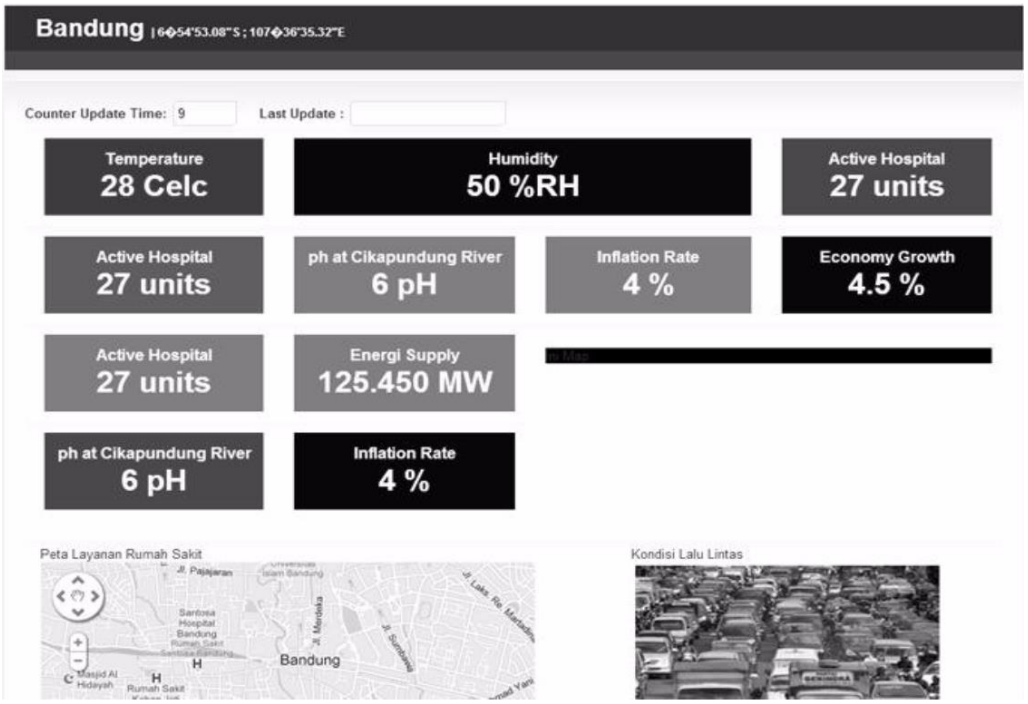


Figura 17 – *Dashboard* exemplo da aplicação. Retirado de (Suakanto et al., 2013).

Como último exemplo deste tipo de arquiteturas, apresenta-se a que foi desenvolvida por (Khan et al., 2013) esta tem como objetivo a análise de dados num modelo *Analytics-as-a-Service*. A arquitetura foi concebida em três camadas seguindo uma abordagem *bottom-up* como ilustra a Figura 18.

É descrito em (Khan et al., 2013) que a primeira camada pretende tratar a aquisição de dados, onde são utilizados os metadados para se lidar com a heterogeneidade, nesta encontramos uma série de repositórios heterogéneos e distribuídos, acedidos através de *web services*, *API* ou conectores desenhados especificamente quando necessário. Utiliza normas disponibilizadas pelo *OGC (Open Geospatial Consortium)* para integrar dados heterogéneos provenientes de sensores.

Refere ainda (Khan et al., 2013) que na segunda camada destacam o uso da *RDF (Resource Description Framework)*, para fazer o mapeamento e a criação de relações tornando os dados semanticamente relevantes e disponíveis para exploração futura, com a possibilidade de descoberta de novos cenários cuja identificação seria impossível em repositórios isolados.

A última camada, diz (Khan et al., 2013), tem como objetivo fornecer a capacidade de explorar os dados, submeter *queries,* criar *workflows,* e disponibilizar algoritmos com o objetivo de extrair informação do repositório.

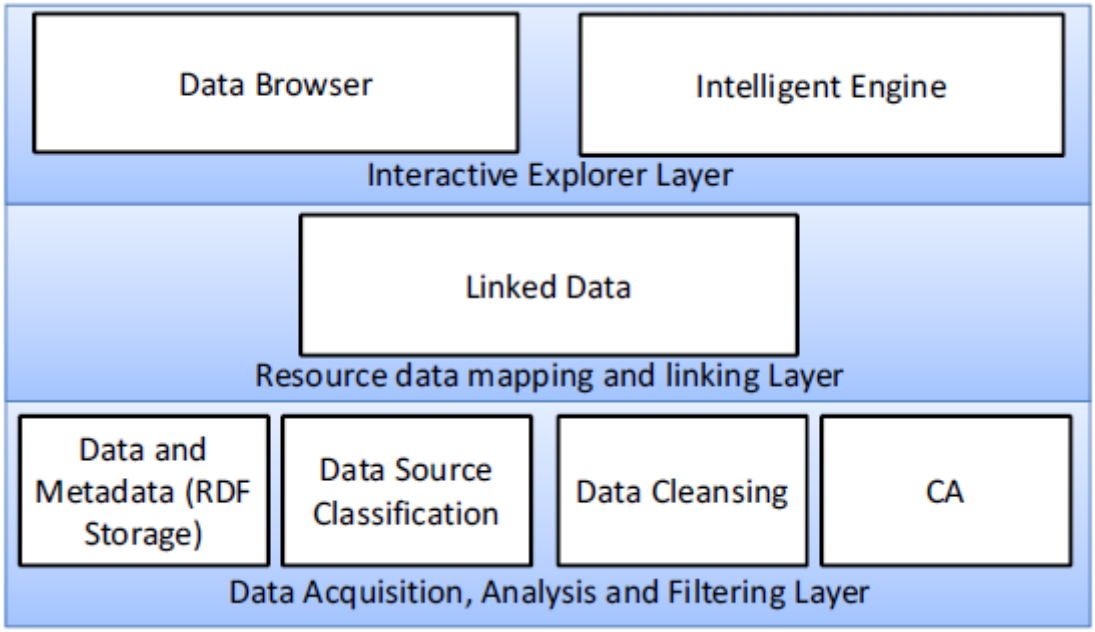


Figura 18 - *Design* da arquitetura proposta. Retirado de (Khan et al., 2013).

De salientar que esta arquitetura foi testada posteriormente em (Khan et al., 2015), na qual os autores implementaram um protótipo usando e demonstrando como uma infraestrutura *cloud* pode ser usada na base analítica através de uma amostra de dados retirada do *Bristol Open Data*, sendo que este protótipo foi implementado em *Hadoop* e em *Spark*.

Como podemos verificar nenhum dos trabalhos apresentados utiliza o conceito *DAaaS* que é proposto neste trabalho, pelo menos do ponto de vista de uma verdadeira infraestrutura que providencie um serviço analítico completamente baseado na *cloud* e que o ofereça no paradigma *as-a-Service,* permitindo a qualquer utilizador fazer as suas próprias análises, podendo ainda contribuir com os seus dados.

3.ENQUADRAMENTO TECNOLÓGICO

Depois de enquadrar conceptualmente os conceitos associados à temática que envolve este trabalho de dissertação, este capítulo visa identificar e descrever as tecnologias que sejam capazes de responder ao desafio proposto, que passa pela implementação de um sistema analítico que possa ser utilizado no paradigma *as-a-Service* no contexto de uma *Smart City.* Para a realização deste objetivo é necessário avaliar as mais variadas propostas que existem no mercado, bem como as suas características, sendo que dois dos principais requisitos a ser tidos em consideração na base de partida é que a solução terá de ser *open source* e *web-based*. Estes requisitos vão de encontro aos princípios que podemos encontrar na arquitetura BASIS, que já foi introduzida no início deste trabalho e que serve de base de enquadramento desta dissertação, sendo que os restantes requisitos serão identificados ao longo deste capitulo, nomeadamente os que advêm da arquitetura proposta em (Costa, 2015). Por conseguinte, segue-se a descrição de quatro plataformas que se enquadram nos requisitos definidos, sendo dada uma visão geral sobre cada uma das plataformas. No final serão apresentadas matrizes que cruzam os diversos aspetos e capacidades de cada uma das plataformas por forma a se percecionar qual será a mais abrangente e que melhor se adapte à realidade que se pretende retratar. Por fim, serão tecidas considerações relativamente à escolha da plataforma que melhor se enquadra.

3.1. Arquitetura BASIS para *Smart Cities*

A proposta que se encontra em (Costa, 2015), tem como principal foco a definição de uma arquitetura capaz de responder aos desafios de uma *Smart City*. Para a sua definição, o autor utilizou uma série de princípios de conceção, nomeadamente:

- Disponibilizar dados de forma aberta, possibilitando o desenvolvimento de novos serviços, não só por iniciativa dos órgãos gestores da *Smart City*, mas também pelos cidadãos e organizações;
- Contemplar várias camadas de abstração, desde a mais conceptual à mais tecnológica;
- Garantir o armazenamento e processamento distribuído dos dados;
- Incluir a noção de segurança, privacidade e confiança dos dados;
- Incorporar formas de gerir o ciclo de vida dos dados, usando para isso conceitos inerentes à partição dos mesmos;

- Estabelecer modos de cooperação entre as pessoas intervenientes no desenvolvimento de serviços de uma *Smart City*;
- Ser *service-oriented* e independente do dispositivo ou plataforma que acede aos dados;
- Fazer uso de tecnologias *open source*, exceto em casos onde a relação custo-benefício justifique outra escolha.

Na Figura 19 podemos observar a camada conceptual da arquitetura BASIS, sendo que o principal componente de relevo neste trabalho é o *Analytics-as-a-Service.* Este está destacado na camada de *Big Data Analytics,* dando especial atenção ao *Analytical-Processing-as-a-Service*, *Visualization-as-a-Service*, e ao Portal *Open Data,* onde os utilizadores interagem com os dados disponíveis.

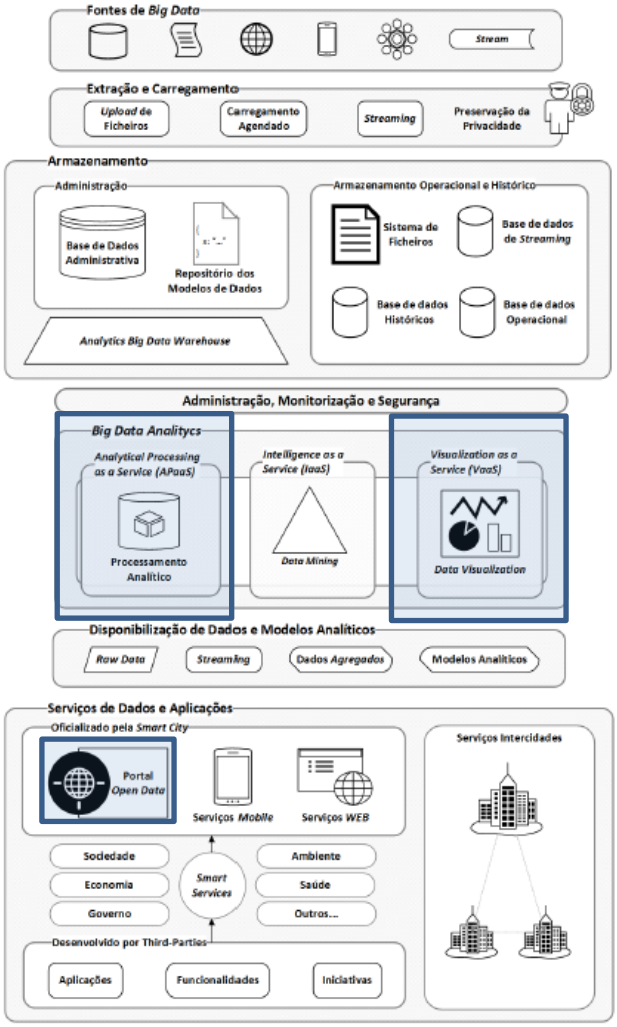


Figura 19 - Camada conceptual da arquitetura BASIS. Retirado de (Costa, 2015).

Nesta arquitetura destaca-se o *Big Data Analytics*, na medida em que o mesmo é responsável pela manipulação e processamento dos dados, incluindo técnicas como *Aata Mining* e *Data Visualization*, de forma a criar serviços que denotem inteligência providenciada pela capacidade de computação atualmente disponível. Nesta componente são designados alguns subcomponentes que conduzem o processo de disponibilização de serviços inteligentes aos cidadãos, desde o pré-processamento dos dados (*Data Cleansing*, transformações, agregações...) até à visualização dos resultados finais, que podem decorrer, ou não, da utilização de técnicas avançadas e algoritmos de *Data Mining*. Um dos princípios de *design* fundamentais, que poderá fomentar a criação de serviços inteligentes por parte da comunidade, é a disponibilização de cada um destes subcomponentes numa ótica "*as-a-Service*", eliminando assim a elevada curva de aprendizagem frequentemente associada a estas tecnologias. Cada um dos componentes de *Big Data Analytics* poderá ser disponibilizado como um serviço no portal *Open Data*, seja através de interação direta, isto é manipulando o conjunto de dados diretamente no portal, ou através de *APIs Web*, com vista a serem utilizadas por aplicações externas, desenvolvidas por terceiros (Costa, 2015).

O autor refere ainda que a utilização da componente de *Big Data Analytics as a Service* no portal *OpenData* permitirá disponibilizar um serviço *Web* de visualizações de dados, provenientes de várias fontes e resultantes dos vários subcomponentes do *Big Data Analytics*, de modo a ser acedido pelos cidadãos para um aumento do conhecimento e envolvimento citadino.

Na Figura 20 conseguimos visualizar a camada tecnológica desta arquitetura, podemos verificar algumas das tecnologias usadas, bem como as restrições ao nível tecnológico, nomeadamente a utilização do *Hive* ou *HDFS*, que devem ser tidos em conta na seleção da plataforma analítica. A definição de papéis para cada camada de armazenamento facilita o papel dos responsáveis pelo *Big Data Analytics,* na medida em que está bem definido em que local está armazenado cada tipo de dados. No que diz respeito ao processamento analítico, esta arquitetura destaca o *Spark* como motor de transformação de dados, na medida em que este tem vindo a aumentar de forma considerável a sua popularidade. No entanto, não restringe este componente ao uso desta tecnologia, mas a outras, desde que sejam adequadas à natureza dos dados a serem processados (Costa, 2015).

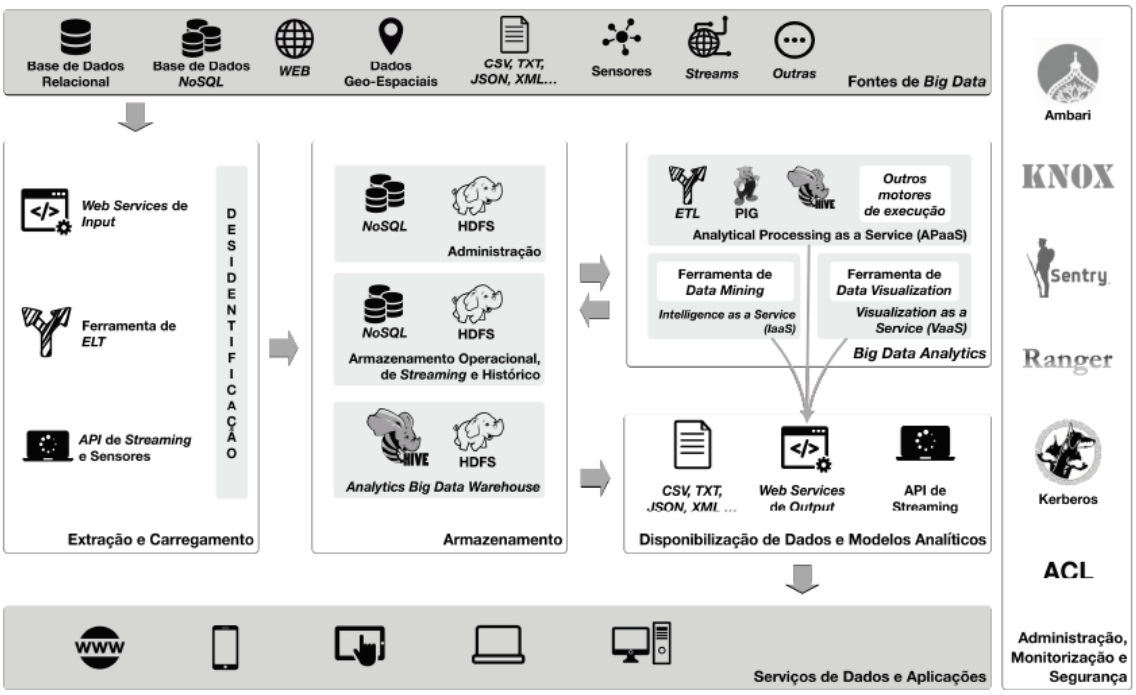


Figura 20 - Camada tecnológica da arquitetura BASIS. Retirado de (Costa, 2015).

Introduzida a arquitetura BASIS, as próximas secções apresentam as diversas tecnologias que poderão permitir a implementação do serviço analítico a desenvolver neste trabalho.

3.2. Plataformas Analíticas

3.2.1. *Pentaho*

A plataforma *Pentaho (Pentaho Inc.)* foi disponibilizada em 2004, oferecendo um serviço analítico ao nível *open source* e está situada na categoria de *Business Intelligence applications*. Neste momento oferece duas modalidades diferentes deste produto, uma *enterprise* e outra *community*, sendo que ambas são classificadas como *open source* mas para se utilizar a versão *enterprise* é necessário adquirir uma licença (*Pentaho*, 2016).

O *Pentaho* oferece um grande espetro de ferramentas analíticas, que vão desde os relatórios básicos até à modelação de análises preditivas, sendo possível analisar e visualizar os dados numa vertente de múltiplas perspectivas (*Pentaho*, 2016). Podemos ver na Figura 21 um exemplo de um *dashboard* criado no *Pentaho*.

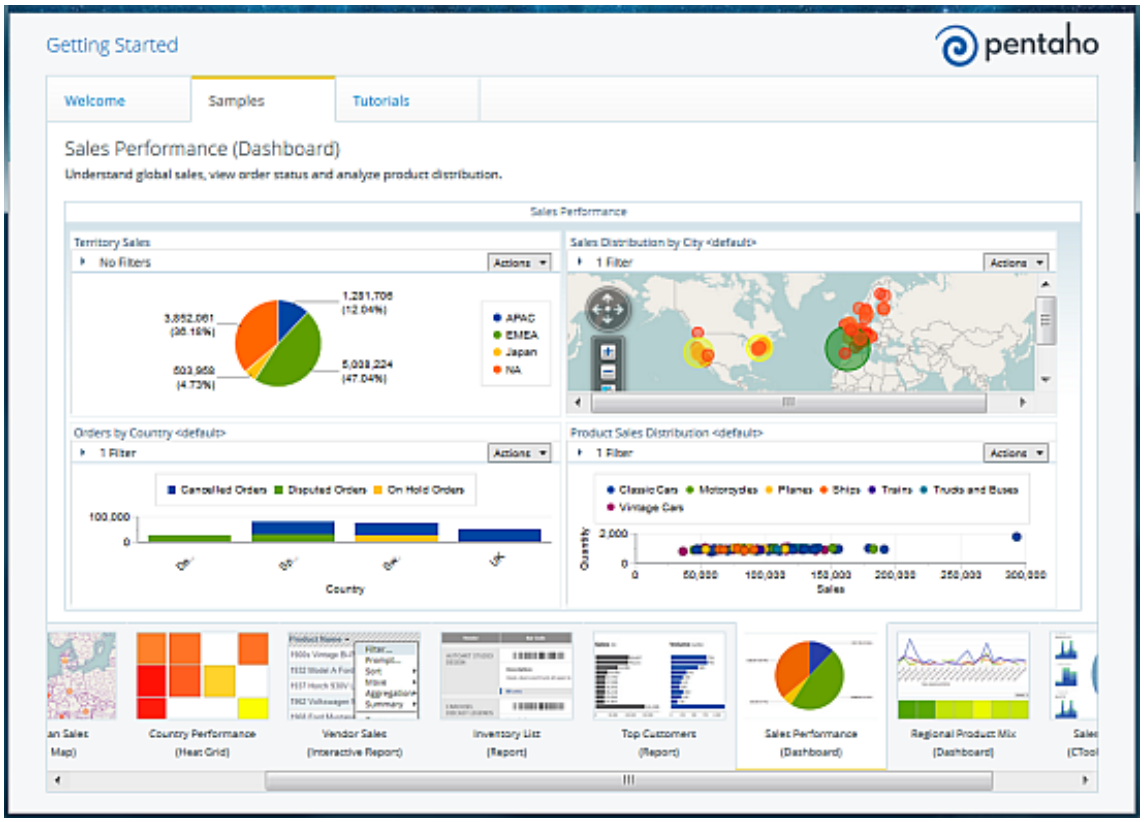


Figura 21 - Exemplo de *dashboard*. Retirado de (*Pentaho*, 2016).

O motor de *BI* do *Pentaho* é uma aplicação *J2EE (Java 2 Platform, Enterprise Edition)* que providencia a plataforma necessária para correr a aplicação baseada na *web*. Uma das características que distingue o *Pentaho* de outras ferramentas de *BI*, é o facto de podermos encontrar diferenças na apresentação dos relatórios impressos com os que são apresentados visualmente na *web* (*Pentaho*, 2016).

Alguns pontos favoráveis desta plataforma, que devem de ser destacados, são o facto de ser considerada intuitiva na medida em que não só os recursos humanos da área das tecnologias de informação, mas também o utilizador comum consegue aceder e visualizar os dados de uma forma simples. Os dados a serem analisados podem ser carregados de um espectro bastante alargado de fontes, desde o Excel até as bases de dados *NoSQL,* como o MongoDB e Cassandra, suportando as últimas distribuições do *Hadoop* nomeadamente a *Cloudera*[3], Hortonworks[4] e MapR[5]. Suporta também o processamento distribuído através de algoritmos de *clustering*. Utiliza técnicas de *caching* para agilizar os relatórios. Estes podem ser exportados em diversos formatos mediante a escolha do utilizador, assim como, a visualização pode facilmente ser filtrada desde uma visão mais generalista dos dados até à mais específica. Permite a integração de aplicações de terceiros como, por exemplo, o *Google Maps.* Os

dispositivos suportados cobrem várias plataformas tais como *android, IOS, web-based* e *Windows* (*Pentaho*, 2016).

Algumas das menos valias encontradas nesta plataforma estão associadas ao conjunto de componentes que a constituem, já que a sua diversidade torna a curva de aprendizagem acentuada inicialmente. A documentação é reduzida e por vezes pouco clara. Não existe sistema de licença perpétua sendo que os direitos de utilização têm de ser adquiridos anualmente (*Pentaho*, 2016).

3.2.2. *SpagoBI*

O *SpagoBI* (Engineering Ingegneria Informativa S.p.A) é uma ferramenta de *BI* 100% *open source*, integrando um leque variado de ferramentas analítica (SpagoBI, 2012). A concretização deste projeto remonta ao ano de 2004, sendo que em 2005 ficou disponível o seu *website* que destacava a visão, *design* e organização do projeto, para dar lugar no início de julho de 2005 a primeira versão do *SpagoBI Suite*. Atualmente podemos encontrar esta plataforma na versão 5.2 (*SpagoBI*, 2016). A sua arquitetura, como podemos ver na Figura 22, é dividida em cinco módulos: *meta,* onde o foco principal são os metadados que podem ser explorados através do ambiente criado pelo *SpagoBI Meta; studio,* onde está o ambiente integrado de desenvolvimento*; sdk,* é a camada de integração entre o *SpagoBI* e outras ferramentas externas; *server,* onde encontramos a plataforma de *BI* que inclui ferramentas analíticas, gestão de segurança e regras, bem como ferramentas de administração; por fim, a camada de *applications,* que contém os modelos analíticos desenvolvidos no *SpagoBI* (SpagoBI, 2014).

Figura 22 - Arquitetura *SpagoBI*. Retirado de (SpagoBI, 2014).

Algumas características consideradas como sendo pontos fortes desta plataforma estão associadas a ser 100% *open source*, onde todos os componentes estão disponíveis sendo que a plataforma, no domínio analítico, conta com mais de vinte motores analíticos que dão a esta

plataforma a capacidade de disponibilizar aos seus utilizadores um alto nível de flexibilidade e a possibilidade de obterem a melhor solução de acordo com as suas necessidades. A sua abordagem é orientada ao utilizador não impondo modelos orientados de acordo com licenciamento (SpagoBI, 2014).

Na Figura 23 podemos ver um exemplo de *dashboard* utilizado por um hospital de São Paulo no Brasil (Klein, 2015).

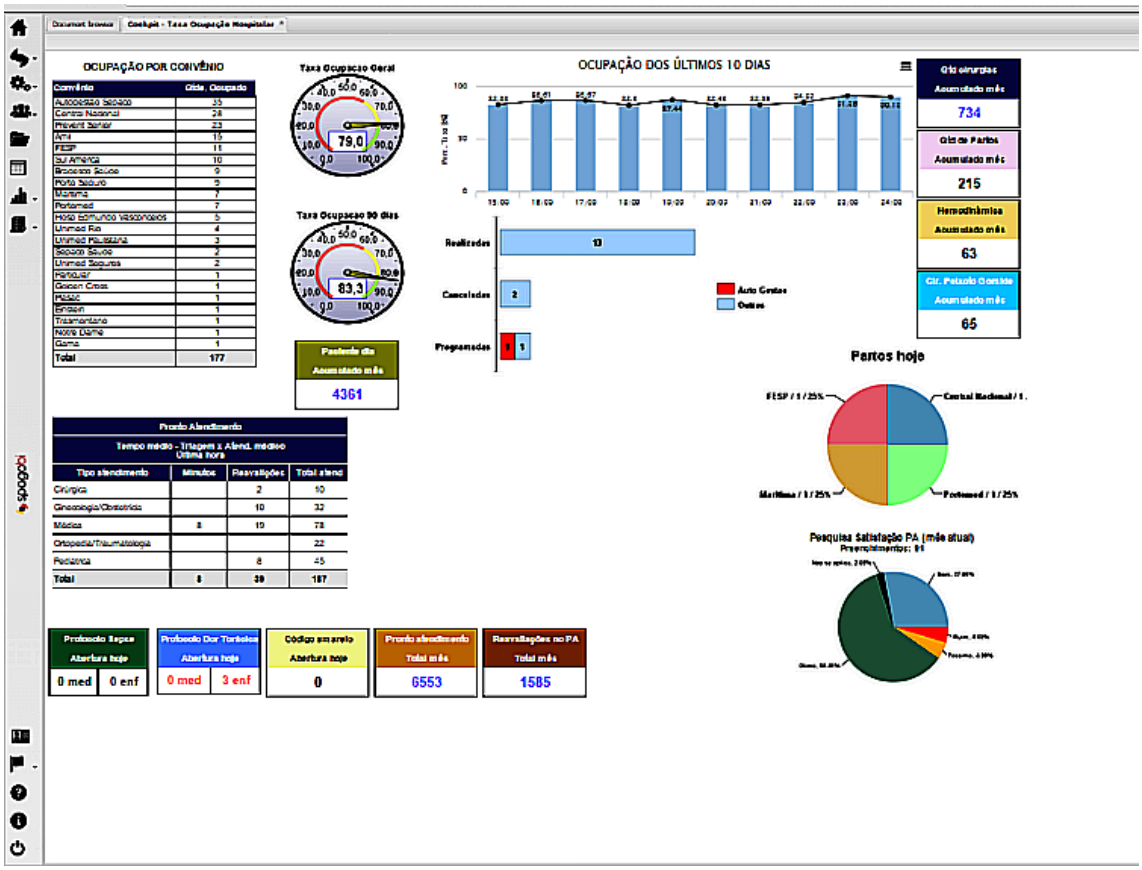


Figura 23 - Exemplo de *dasboard* usado pelo hospital de São Paulo, Brasil. Retirado de (Klein, 2015).

No que diz respeito ao *Big Data* a plataforma da *Spago BI* está preparada para trabalhar com grandes volumes de dados heterogéneos. Para tal permite a utilização de tecnologias *NoSQL* como *Hive*, *Cassandra, OrientDB, MongoDB*, assim como dá suporte para as distribuições do *Hadoop*, como *Hontonworks* e *Cloudera* (SpagoBI, 2014).

3.2.3. *BIRT*

O projeto *BIRT* (*Business Intelligence and Reporting Tools*) (Eclipse Inc.) nasceu em 2004 e foi evoluindo até aos dias de hoje em que se encontra na versão 4.5. Este é um projeto *open source* que visa fornecer as capacidades necessárias à criação de aplicativos *web-based*

especialmente os baseados em Java e Java EE. *BIRT* é um projeto de *software* disponibilizado pela fundação Eclipse[6].

A plataforma é constituída por dois componentes, um deles é disponibilizado para que em conjunto com o Eclipse IDE seja possível criar relatórios, o outro componente permite integrar os relatórios criados no *BIRT* em uma aplicação java. O *BIRT* permite a ligação ao *Hadoop*, com o objetivo de visualização e tratamento dos dados através do *Hive* usando o *HQL (Hive Query Language)*. O *Hive* é uma infraestrutura disponível no ecossistema do *Hadoop* que disponibiliza sumários, possibilita *queries* e análises (Dodson, 2013).

Na Figura 24 podemos observar um exemplo de um ambiente analítico criado com o *BIRT*.

Figura 24 - Exemplo de ambiente analítico criados através do *BIRT*. Retirado de (Eclipse, 2016).

3.2.4. *Jaspersoft*

A plataforma *Jaspersoft* (TIBCO) foi disponibilizada em 2001, sendo que neste momento está na versão 6.2 desde dezembro de 2015. A *Jaspersoft* tem disponibilizado em versão *comunity* mas também uma versão comercializada pela empresa TIBCO que adquiriu a *Jaspersoft* em 2014 (Jaspersoft, 2015a). Como podemos observar na Figura 25 esta plataforma

suporta diversas tecnologias como *J2EE*, MongoDB entre outros, apesar de não estar na imagem permite ainda, através de conectores disponíveis a partir da versão 5.5, a ligação ao *Hadoop* (Jaspersoft, 2015b).

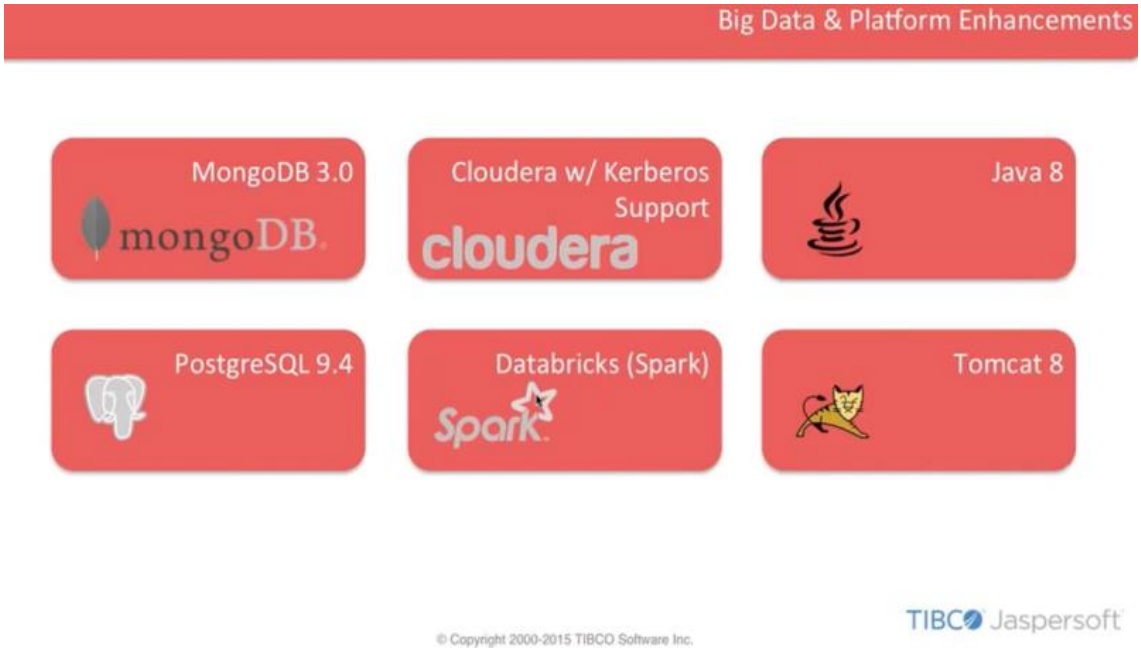


Figura 25 – Técnologias de suporte *Jaspersoft*. Retirado de (Jaspersoft, 2015a).

Apresenta-se como uma plataforma única de arquitetura modular, sendo 100% baseada em java obedecendo aos *standards J2EE*, assim como permite a utilização de tecnologias como *XML (Extensible Markup Language)*, *HTTP (Hypertext Transfer Protocol)*, *JSP (Java Server Pages)*, *SOA (Service Oriented Architecture) web services* e integração com *CSS (Cascading Style Sheets)*. É também disponibilizado com suporte em várias línguas (Jaspersoft, 2015a). Na Figura 26 podemos visualizar um exemplo de um *dashboard* criado na plataforma *Jaspersoft*.

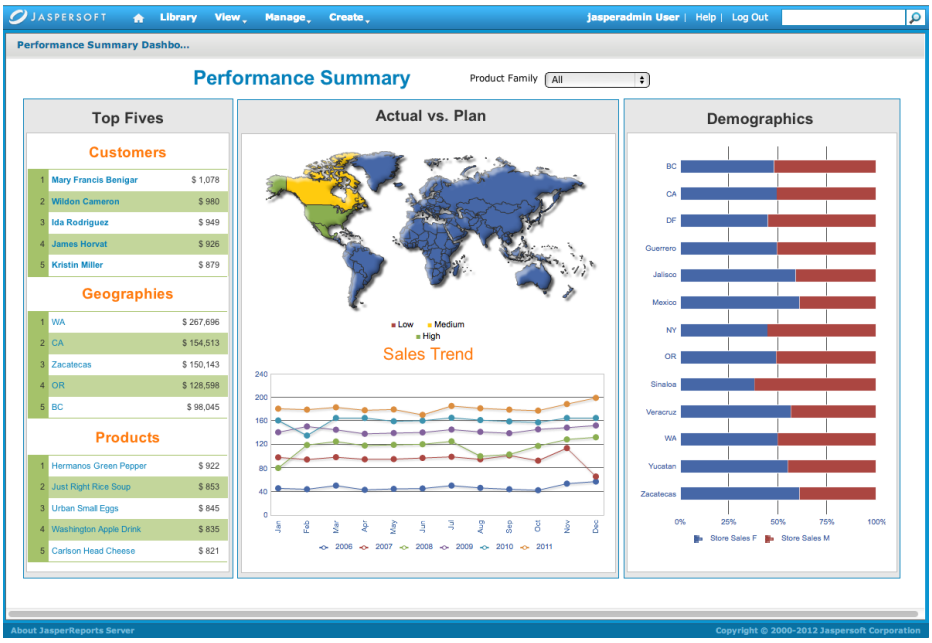


Figura 26 – Exemplo de *dashboard* criado com *Jaspersoft*. Retirado de (Jaspersoft, 2015a).

3.2.5. Análise comparativa entre plataformas

Depois de brevemente descritas cada uma das plataformas, existe a necessidade de explorar cada uma das suas diferentes vertentes. A análise feita às plataformas foi baseada nos sites oficiais de cada uma delas, bem como em contacto direto através de *e-mail* com o suporte. Com esta abordagem foi possível elaborar um conjunto de tabelas que descrevem as principais funcionalidades de cada uma delas.

De forma a facilitar a análise comparativa, é utilizada a seguinte legenda para classificação das características de cada ferramenta:

Legenda:
Sim - ✔
Não - ✘
Não aplicável – N/A

Tabela 1 - Informações gerais.

Informação Gerais	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Web site open source	http://community.Pentaho.com/projects/reporting/	http://www.SpagoBI.org/	http://www.eclipse.org/BIRT/	http://community.Jaspersoft.com/
Web site comercial	http://www.Pentaho.com/	N/A	http://developer.actuate.com/	http://Jaspersoft.com/reporting-software
Licença	*Pentaho Reporting LGPL V2.1* (ou anterior)	*Mozilla Public Licence v. 2.0*	*Eclipse Public License*	*JasperReports Lib LGPLV3* e *Jaspersoft Studio EPL*
Desenvolvimento de relatórios	*Pentaho Report Designer*	*SpagoBI Studio* e *Built on Eclipse*	*BIRT Report Designer* e *Built on Eclipse*	*Jaspersoft Studio* e *Built on Eclipse*
Plataforma de instalação	*Windows, Linux, Mac OS X*	*Windows, Linux, Mac OS X*	*Windows, Linux, Mac OS X*	*Windows, Linux, Mac OS X*
Plug-in disponível para *Eclipse*	✗	✓	✓	✓
Plug-in disponível para *NetBeans*	✗	✗	✗	✗
Cliente *Java*	✓	✓	✓	✓
Paradigma de desenvolvimento	*Banded reports pixel positioning*	*Web page design frames tables lists*	*Web page design frames tables lists*	*Banded reports pixel positioning*
Formato de relatórios	*XML, PRTP*	*RTP Design Jasper JRXML*	*XML, RPTDESIGN*	*JRXML, JASPER*
Multiutilizador	✓	✓	✓ na versão comercial	✓

No que diz respeito à Tabela 1, destacamos apenas dois pontos, um deles é que a plataforma *SpagoBI* é a única 100% livre, o que é um facto positivo pois as versões comerciais tendem a evoluir de uma forma em que a versão não paga fica sem as melhores funcionalidades, no entanto, as outras plataformas estão associadas a grandes companhias, o que dá estabilidade nomeadamente na continuidade das plataformas. O segundo aspeto a destacar está associado ao paradigma de desenvolvimento "*Banded reports pixel positioning*" utilizado pelo *Pentaho* e pela *Jaspersoft*, que está mais direcionado aos relatórios para

impressão e não tão voltado para os *layouts web* dinâmicos em que o *SpagoBI* e o *BIRT* utilizam "*Web page design frames tables lists*" sendo este mais direcionado para a *web*.

Tabela 2 - Componente de desenvolvimento de relatórios.

Componente de desenvolvimento de relatórios	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Desenvolvimento de relatórios base	✓	✓	✓	✓
Formas geométricas	✓	✓	✓	✓
Código de barras	✓	✗	✓	✓
Notas em modo de *design*	✗	✓ como elemento do relatório	✓ no editor de propriedades	✓ como elemento do relatório
Tabela de conteúdos como componente base	✓	✓	✓	✓
Sub-relatórios	✓	✓	✓	✓
Componente de relatórios lado a lado	✓	✓	✓	✓
Tabelas	✓	✓	✓	✓
Cross tabs	✓ versão experimental	✓	✓	✓
Planeamento horizontal	✗	✓ com tabela	✓	✗
Layout multicoluna	✗	✓	✗	✓
Ações em gráficos	✓	✓	✓	✓
Controlo em *CSS*	✓	✓	✓	✓
Formatação por fórmula	✓	✓	✓	✓

Pelas características apresentadas na Tabela 2 os relatórios construídos pela *Pentaho* demonstram um caris mais simples e moderado, as plataformas *BIRT* e *Jaspersoft* conseguem ter mais abrangência e são mais indicadas para relatórios mais complexos. No entanto, quem se destaca pela positiva neste ponto é a *SpagoBI* sendo a mais completa tendo todas as funcionalidades analisadas disponíveis.

Tabela 3 - Origem de dados.

Origem de dados	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Múltiplos repositórios de dados e *queries* por relatório	✓ Via sub-relatório / gráfico	✓	✓	✓ Via sub-relatório / gráfico
Suporte de união de múltiplos repositórios de dados através do módulo de *design*	✓	✗ Será disponibilizado na próxima versão	✓	✗
Possibilidade de manipular a *query* no relatório	✓	✓	✓	✓ Parcial
Origens de dados não *JDBC (Java Database Connectivity)*				
Cassandra	✓	✓	✓	✓
CSV	✓	✓	✓	✓
Excel	✓	✓	✓	✓
Hadoop Hive	✓	✓	✓	✓
HBase	✓	✓	✓	✓
Hibernate	✗	✓	✓	✓
JNDI	✓	✓	✓	✓
JSON (JavaScript Object Notation)	✓	✓	✓	✓
MongoDB	✓	✓	✓	✓
Script	*BeanShell* *Groovy* *JACL* *JavaScript*	*JavaScript Groovy*	*JavaScript*	*JavaScript Groovy*

Origem de dados	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
	Jython *Netrexx* *XSLT*			
Web services	✓	✓	✓	✓
XML	✓	✓	✓	✓
JDBC Drivers	*Firebird SQL* *Greenplum* *Gupta SQL Base* *H2* *Hadoop Hive* *Hadoop Hive 2* *Hypersonic*	*JDBC genérico* *HSQLDB* *MySQL* *Oracle* *MariaDB* *PostgreSQL* *Teradata*	*JDBC genérico*	*JDBC genérico* *Cloudscape* *Derby* *Firebird* *Hadoop Hive* *H2* *HSQLDB*
	IBM DB2 *Infobright* *Informix* *Ingres* *Ingres VectorWise* *Intersystems Cache* *Kettle thin JDBC Driver* *KingbaseES*	*MS SQL Server*		*IBM DB2* *Inetdae7* *Informix* *Ingres* *JDBC - ODBC Bridge* *MS SQLServer* *MariaDB* *Mondrian*

Origem de dados	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
	LucidDB *MS Access* *MS SQLServer* *MS SQL Server (Native)* *MaxDB (SAP DB)* *MonetDB* *MySQL* *Native Mondrian* *Neoview* *Netezza* *Oracle* *Oracle RDB* *PostgreSQL* *Remedy Action Request System* *SAP ERP System* *SQLite* *Sybase* *SybaseIQ* *Teradata* *UniVerse database* *Vertica 5+* *dbase III/IV/5*			*MySQL* *OLAP4J* *Oracle* *PostgreSQL* *SQLite* *Sybase* *Vertica*

	Pentaho	*SpagoBI*	*BIRT*	*Jaspersoft*
Criação de *queries* em ambiente gráfico	✓	✓	✓	✓
Programação por *script*	*JavaScript* *BSF (Bean Script Framework)* *BSH (Bean-Script Host)* *Single Value Query* *Metadata data-source scripting extension*	*Groovy*	*JavaScript* *Java Event Handlers*	*JavaScript* *Groovy* *Java*

No que diz respeito à Tabela 3, podemos destacar que o *Pentaho* tem uma excelente cobertura ao nível de *drivers JDBC* em que é a plataforma que tem mais cobertura ao nível das plataformas *NoSQL*, deixando apenas o Hibernate de fora. A *SpagoBI* e a *Jaspersoft* demonstram que o suporte de união de múltiplos repositórios de dados através do módulo de *design* não é possível, pelo que a *SpagoBI* prevê que este fique disponível na próxima versão. Nesta área podemos ainda notar que o *BIRT* apenas não se destaca nos *drivers JDBC*, pelo que, apenas é possível ficarem disponíveis pela instalação manual no interface do utilizador.

Tabela 4 - Formato de exportação.

Formato de exportação	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
XHTML	✓	✗	✓	✓
PDF	✓	✓	✓	✓
XLS e XLSX	✓	✓	✓	✓
XML	✓ via *API*	✗	✓	✓
Plain Text	✓	✗	✓	✓
Rich Text	✓	✗	✓	✓

Formato de exportação	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Rich Text	✓	✗	✓	✓
Powerpoint	✗	✓	✓	✓
CSV	✓	✓	✓	✓
Postscript	✗	✗	✓	✓
OpenOffice	✗	✗	✓	✓
Flash	✗	✗	✗	✓

No que diz respeito à Tabela 4, formatos de exportação, podemos ver que o *Jaspersoft* e o *BIRT* são os mais abrangentes, de seguida temos o *Pentaho* e por fim o *SpagoBI* como sendo a plataforma mais incompleta neste ponto. No entanto todas conseguem exportar para os principais formatos utilizados, nomeadamente *pdf, xls, xlsx e csv*.

Tabela 5 - Gráficos.

Gráficos	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Wizard	✗	✗	✓	✓
Interativos	✗	✓ Passar o rato, dicas, etc.	✓ Passar o rato, dicas, etc.	✓ *Hyperlinks*
Temas	✗	✓	✓	✓
Controlo total dos elementos	✗	✓	✓	✓
Vários tipos (2D, 3D, barras, linhas, etc.)	✓	✓	✓	✓
Anel	✓	✗	✓	✗
Séries de tempo	✗	✓	✓	✓
Diferenciadores	✓	✗	✓	✗
Termómetro	✓	✓	✗	✓

Gráficos	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Gantt	✗	✓	✓	✓
Componente de Mapas	✗	✗	✗	✓
Gráficos	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Vetores	✗	✓	✓	✓

Da Tabela 5, podemos concluir que a *Pentaho* é a que menos se destaca e a que mais se destaca é a *Jaspersoft* contendo nativamente a componente de mapas.

Tabela 6 - Parametrização de relatórios.

Parametrização de relatórios	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Listas codificadas de valores	✓	✓	✓	✓
Parâmetros dinâmicos	✓	✓	✓	✓
Calendário	✓	✓	✓	✓
Valores pré-definidos	✓	✓	✓	✓
Caixas *drop-down*	✓	✓	✓	✓
Botões rádio	✓	✓	✓	✓
Caixas de *check*	✓	✓	✓	✓
Caixas de combinação	✓	✓	✓	✓

Da Tabela 6, podemos concluir que todas as plataformas na sua generalidade permitem parametrizar os relatórios de uma forma uniforme entre elas.

Tabela 7 - Agregações – sumário de dados.

Agregações – sumário de dados	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Agregações comuns	*Average* *Count* *Count by Page* *Group Count* *Sum* *Minimum* *Maximum* *Sum Quotient* *Sum Quotient Percent*	*Average* *Concatenate* *Count* *Sum* *Distinct* *First* *Irr* *Is-bottom-n* *Is-bottom-npercent* *Is-top-n*	*Average* *Count* *Distinct Count* *First* *Is-Bottom-N* *Is-Botton-N-Percent* *Is-Top-N* *Is-Top-N-Percent* *Last*	*Average* *Count* *Distinct Count* *Sum* *First* *Lowest (Minimum)* *Highest (Maximum)* *Standard Deviation* *Variance*
Agregações comuns	*Calculation* *Count for Page* *Sum for Page* *Sum (Running)* *Count (Running)* *Group Count (Running)* *Count Distinct (Running)* *Average (Running)* *Minimum (Running)* *Maximum (Running)* *Percent of Total (Running)*	*is-toppercent* *Last* *Max* *Median* *Min* *Mirr* *Mode* *Movingave* *Npv* *Percentile* *Percentrank* *Percentsum* *Quartile* *Rank* *Runningcou* *Nt* *Runningnpv* *Runningsum* *Stddev*	*Max* *Median* *Min* *Mode* *Moving Ave* *Percentile* *Percent-Rank* *Percent-Sum* *Quartile* *Rank* *Running Count* *Running Sum* *Standard Deviation* *Sum* *Variance* *Weighted Average*	*System*

Agregações – sumário de dados	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Definição de funções/expressões	✓	✓ *Java, Javascript* ou *Groovy*	✓ *Java, JavaScript*	✓ *Java, Javascript* ou *Groovy*
Utilizador define agregações	✓	✓	✓	✓

Da Tabela 7, podemos verificar que todas as plataformas têm as principais formas de agregação de dados, sendo que a *SpagoBI* consegue ter maior variedade e a *Jaspersoft* menor variedade.

Tabela 8 - Reutilização de componentes.

Reutilização de componentes	*Pentaho*	*SpagoBI*	*BIRT*	*Jaspersoft*
Modelos	✓	✓	✓	✓
Definição de librarias	✓ Não refinado	✓	✓	✓ Não refinado
Estilos, cores, fontes,	✓	✓	✓	✓
CSS	✓	✓	✓	✓

Por último, em relação à Tabela 8, reutilização de componentes, podemos constatar que todas as plataformas, de uma forma uniforme, correspondem sem se diferenciarem neste ponto.

Depois de analisar todas as tabelas podemos pois, tecer algumas considerações sobre cada uma das plataformas de forma a sintetizar a informação recolhida e assim poder tirar conclusões.

Feita esta avaliação, a proposta será a utilização da plataforma *SpagoBI* como plataforma base na medida em que esta, como podemos verificar, é abrangente e destaca-se das demais em vários aspetos, nomeadamente no que diz respeito a ser a única 100% *open source*. O *Pentaho* apresenta características em que a necessidade de relatórios simples é perfeitamente colmatada. O *BIRT* é focado nativamente em relatórios e não propriamente em análises, sendo que a versão não comercial não contempla segurança, gestão de utilizadores, repositório de relatórios, sendo que as outras três opções contemplam estas funcionalidades. O *Jaspersoft* mostra-se uma boa plataforma, apesar de ser necessária a compilação dos relatórios como passo adicional.

3.3. Experimentação da plataforma *SpagoBI*

Nesta secção será apresentada a sistematização da infraestrutura *SpagoBI,* incluindo a execução de testes a algumas das funcionalidades que se encontram por defeito na plataforma. Esta execução tem como principal objetivo perceber como estas funcionalidades poderão ser usadas no serviço analítico que se pretende desenvolver. Neste contexto será ainda testada a tecnologia de armazenamento em *HDFS* juntamente com *Hive* como aconselhado em (Costa, 2015), com o objetivo de perceber o comportamento do *SpagoBI* com este tipo de tecnologias. Na subsecção 3.3.1 começamos por detalhar a arquitetura do servidor da *SpagoBI*, descrevendo os seus principais componentes de uma forma detalhada. De seguida, na subsecção 3.3.2, é apresentada a arquitetura tecnológica usada para o ambiente de teste seguido da descrição das principais funcionalidades do *SpagoBI* através de vários cenários práticos.

3.3.1. Características genéricas da plataforma

Como verificado na secção 3.2, a plataforma *SpagoBI* revela-se a melhor opção disponível. Como a análise comparativa demonstrou na subsecção 3.2.5, abrange as principais necessidades encontradas para a resolução do problema proposto, no entanto existe a necessidade de testar tecnologicamente a plataforma. Em primeiro lugar, e como é visível na Figura 27, aprofundamos a arquitetura do componente *server* da plataforma, que tem como finalidade disponibilizar o serviço analítico.

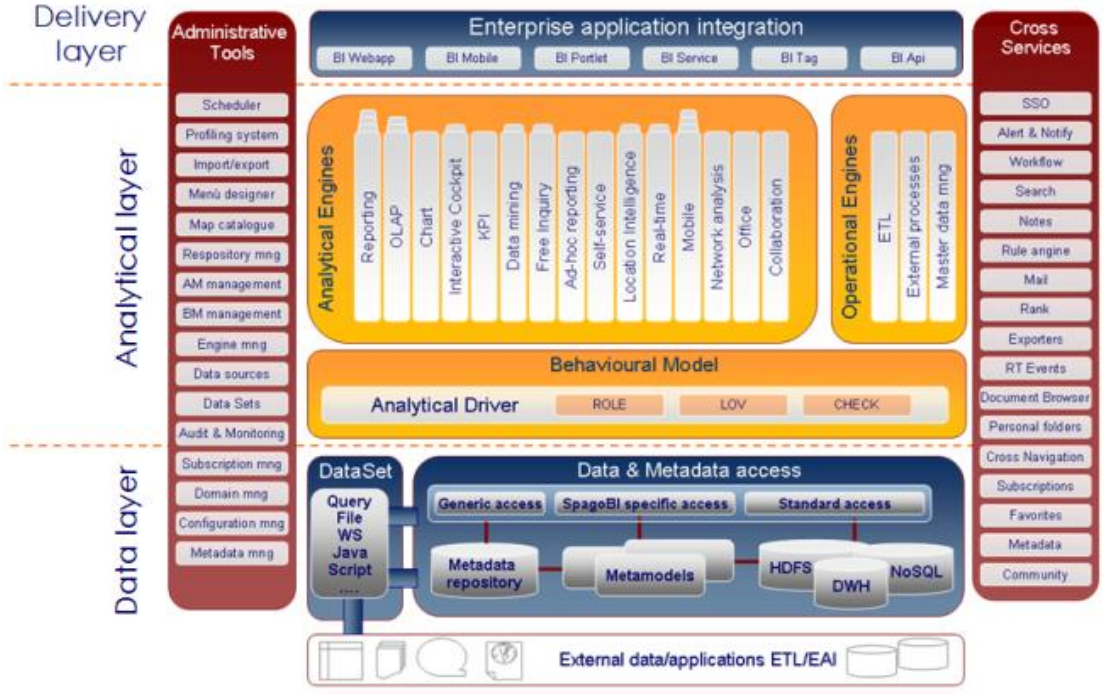


Figura 27 - Arquitetura do servidor *SpagoBI*. Retirado de (Bernabei, 2014).

Sucintamente, o modelo analítico é o elemento chave do *SpagoBI*, este abrange uma vasta área de necessidades analíticas providenciando diversas soluções tais como (Bernabei, 2014):

- *Reporting* – Onde é permitida a realização de relatórios e exportação dos mesmos em diversos formatos, bem como a sua criação de uma forma livre através do *Ad-hoc reporting*;
- *OLAP* – Permite a interação com modelos multidimensionais de uma forma flexível dando acesso a motores de *OLAP*;
- *Charts* – Onde é possível a criação de gráficos, de forma interativa através de *widgets;*
- *KPI (Key Performance Indicator)* – Onde são criados, geridos e visualizados os modelos hierárquicos;
- *Dashboards* – Estes são gerados de uma forma interativa e intuitiva, agregando diversos tipos de documentos, como gráficos e tabelas de diversos tipos em uma única visualização;
- *Location intelligence* – Utilizado para conexões em tempo de execução entre dados geográficos e os dados do negócio;
- *Free inquiry* – Onde é possível a criação de consultas por meio de ferramentas totalmente gráficas;
- *Data Mining* – Permite descobrir padrões de informação escondida entre uma grande quantidade de dados;
- *Real-time* – Onde são produzidas visualizações para a monitorização em tempo real;
- *Collaboration* – Para a criação automática de pastas de relatórios organizados, com comentários e notas;
- *Office automation* – Para a publicação de documentos pessoais em ambientes de *BI*;
- *ETL* – Existe a possibilidade de carregar os dados para o *Data Warehouse* e geri-lo;
- *Mobile* – Com base nos dispositivos de interação no paradigma *touch-screen*, concebido para uma eficiente capacidade de trabalho *off-line;*
- *External processes* – Permite a gestão de processos personalizados, funcionando no fundo e/ou a partir de uma hora programada;

- *Master data management* – Onde se pode tirar proveito das funcionalidades *write-back* das bases de dados;
- *Network analysis* – Permite aos utilizadores visualizar e interpretar as relações entre um conjunto de entidades.

A camada *Behavioural Model* regula a visibilidade dos diversos documentos de acordo com as regras estabelecidas para cada tipo de utilizador. Os documentos analíticos relacionados com este componente guiam o comportamento de acordo com as regras estabelecidas. Este permite (Bernabei, 2014):

- Reduzir o número de documentos analíticos necessários;
- Codificar univocamente as regras que regulam o comportamento;
- Garantir a uniformidade do crescimento do projeto ao longo do tempo;
- Garantir o respeito das regras ao longo do tempo, sem qualquer tipo de limite em todos os motores, assim como, dos documentos analíticos que o utilizador vai adicionando.

A camada *Administration Tools* serve de suporte aos programadores, na medida em que, permite a realização de testes e a administração do dia-a-dia, providenciando diversas funcionalidades assim como (Bernabei, 2014):

- Agendamentos;
- Sincronização de regras;
- Perfis de utilizadores;
- Importar e exportar;
- Gestão de menus;
- Catálogo de mapas;
- Gestão do repositório de documentos, modelos analíticos, modelos de comportamento;
- Configuração de motores e origem de dados;
- Auditoria e monitorização;
- Gestão de subscrições;
- Gerir metadados do negócio.

A camada *Cross Services* inclui uma série de funcionalidades que podem ser utilizadas em todas as áreas analíticas (Bernabei, 2014):

- *Single Sign On;*
- Notificações e alertas;

- *Workflow;*
- Motor de pesquisa;
- Ferramentas de colaboração;
- Motor de regras;
- Envio de *e-mails*;
- *Ranking*;
- Exportação em vários formatos;
- Eventos em tempo real;
- Navegação documental;
- Ficheiros pessoais;
- Navegação cruzada;
- Subscrições;
- Visualização de metadados.

3.3.2. Principais funcionalidades

Descrita a plataforma sobre o ponto de vista da sua arquitetura e seus diversos componentes, importa agora testar a plataforma sobre o ponto de vista tecnológico. Para tal foi necessário instalar a infraestrutura base, ilustrada na Figura 28, que representa tecnologicamente a abordagem usada para os diversos testes na plataforma *SpagoBI*.

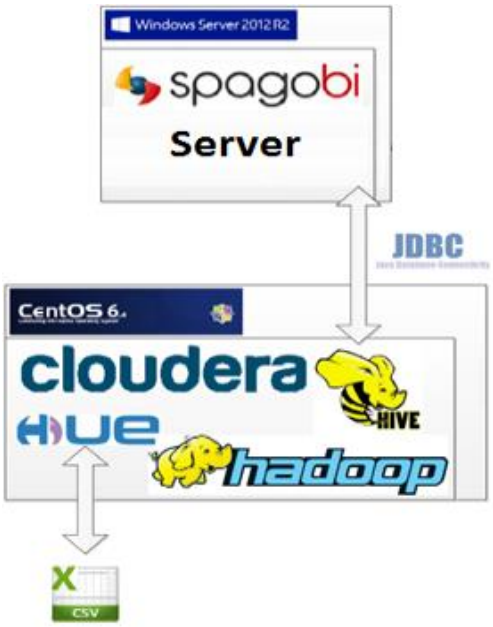


Figura 28 - Arquitetura base de teste da plataforma *SpagoBI.*

Numa primeira fase, a versão testada foi "*All in One SpagoBI 5.1*" instalada em uma máquina virtual com o *Windows Server 2012 R2 Standard*. Foi ainda necessário encontrar uma

plataforma de *Big Data* que já tivesse a infraestrutura base instalada, nomeadamente o *Hadoop* e o *Hive,* e que abstraísse a complexidade de instalação de raiz deste tipo de sistema, proporcionando assim um ambiente de teste considerado adequado. Para tal foi escolhida a máquina virtual da *Cloudera* versão 5.5, onde os componentes necessários já estão instalados e funcionais. Com estas duas máquinas virtuais temos então a nossa base de teste.

Devido à falta de capacidade computacional foi criada uma conta sem custos na plataforma do Google. No entanto, em tempo útil, não foi possível implementar toda a infraestrutura de teste, tanto pela complexidade da mesma do ponto de vista de instalação, visto que não é possível fazer *upload* de uma forma direta de uma máquina virtual para a *cloud,* nomeadamente a máquina da *Cloudera*, sem antes alterar diversos parâmetros na base do ficheiro do disco, quer pelas limitações de uma conta sem custos no que diz respeito aos recursos disponíveis. Para colmatar esta falta de recursos, a Universidade disponibilizou uma máquina com os recursos mínimos necessários para que as duas máquinas virtuais pudessem ser executadas em simultâneo. A máquina *SpagoBI* foi instalada com dois cores e 4GB de *RAM (Random Access Memory)*, enquanto a máquina *Cloudera* foi instalada com 4 cores e 8GB de *RAM*.

A plataforma *SpagoBI* ficou disponível no link 193.136.11.165:8080/SpagoBI. Genericamente, e como é visível na Figura 29, por defeito, a plataforma está dividida por três utilizadores, *administrator*, com acesso total a todas as funcionalidades da plataforma, *showcase user* e o *business user,* cujo acesso é parcial e limitado, servindo apenas para demonstração de ambientes restritos.

Figura 29 - Página inicial da plataforma *SpagoBI.*

Com o utilizador administrador, e como visível na Figura 30, menu do lado esquerdo, temos as diversas funcionalidades: página principal, onde e a qualquer momento da navegação se pode voltar à página principal da plataforma; menu do utilizador, onde os gráficos, relatórios, *dashboards* e todos os documentos guardados pelo utilizador residem; recursos, onde são definidas as fontes de dados, geridos os catálogos de dados e as definições do servidor; perfis, onde são geridas as permissões, regras, modelos de comportamento, configurações de menu e funcionalidades; documentos, onde podem ser criados os *dashboards* e documentos genéricos e acedidas as análises georreferenciadas; modelos de dados, onde estão disponíveis os modelos de dados que podem ser analisados; *KPI*, criação e a modelação de *KPI* bem como a criação e definição de alarmes; gestão do repositório, importação e exportação de modelos de dados e de documentos criados, tarefas agendadas e análises em redes sociais; línguas, contém as várias línguas que a plataforma permite; ajuda, direciona para o portal *wiki* da *SpagoBI*; info, contém os créditos; saída, para fazer a saída do utilizador da plataforma.

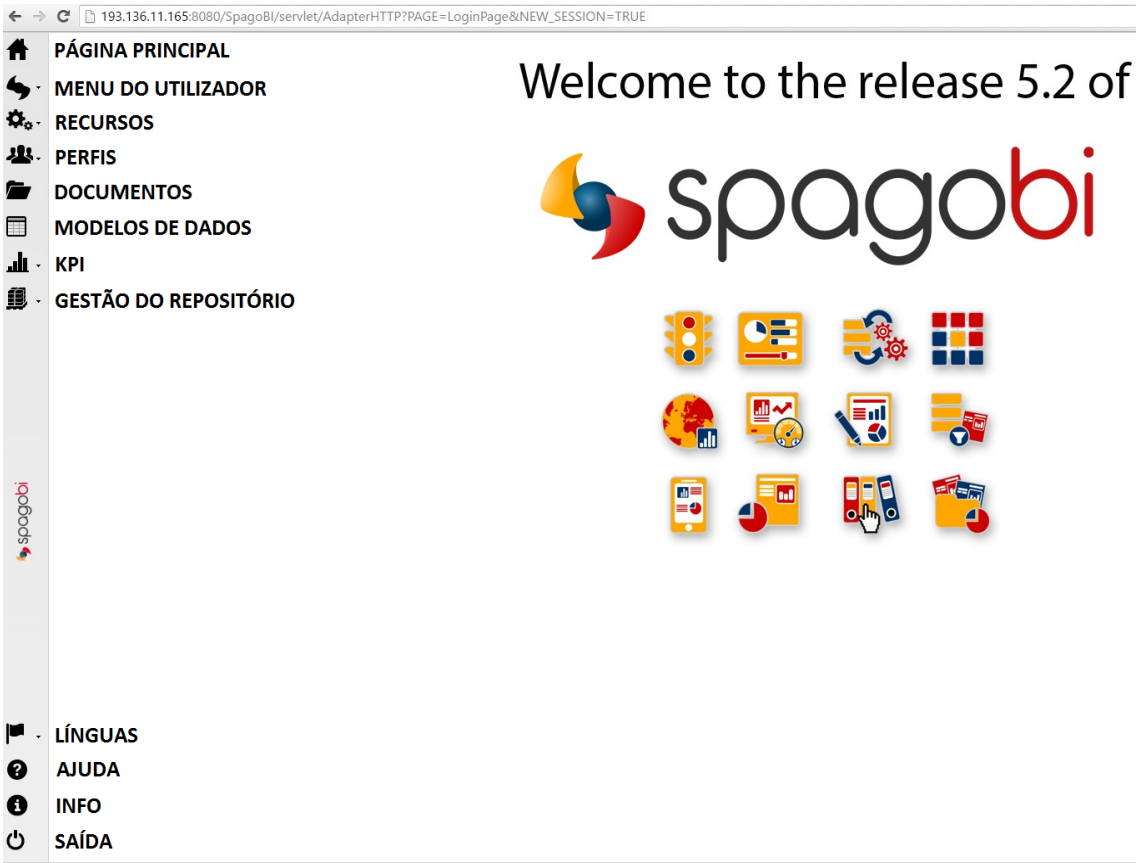


Figura 30 - Ambiente de administrador da plataforma *SpagoBI*.

Como sugere a arquitetura base exposta em (Costa, 2015) é necessário perceber se a plataforma tem a capacidade de análise através de tecnologias como o *Hive*

Na exploração desta abordagem foi levantada a questão de como realizar a conexão entre o *SpagoBI* e a *Cloudera* através do motor *Hive*, visto que nativamente o *SpagoBI* não contém *drivers* de conexão. Para a resolução desta questão foram tidas em consideração duas abordagens, uma de contacto direto com *SpagoBI* através de *e-mail*, e outra através de pesquisa na própria base de conhecimento da *SpagoBI*. Com estas abordagens a questão foi resolvida através de ficheiros de conexão *JDBC* integrados na plataforma da *SpagoBI*, disponibilizados pelo suporte técnico e através da pesquisa na comunidade e wiki da *SpagoBI*, foi possível estabelecer a conexão com sucesso.

Com o objetivo de analisar dados disponíveis em tabelas no *Hive,* foi integrado através do ambiente *Hue*[6], que é uma interface *web* que permite a utilização do ecossistema *Hadoop* de uma forma fácil, dados pessoais de consumos elétricos e produção fotovoltaica. Estes dados deram origem a tabelas no *Hive*. Através da conexão estabelecida entre o *SpagoBI* e o *Cloudera*, foram construídos dois *dashboards* de análise na plataforma *SpagoBI*.

Como é visível na Figura 31, foi elaborado um *dashboard* que permite a análise de consumos elétricos de uma forma dinâmica e simples.

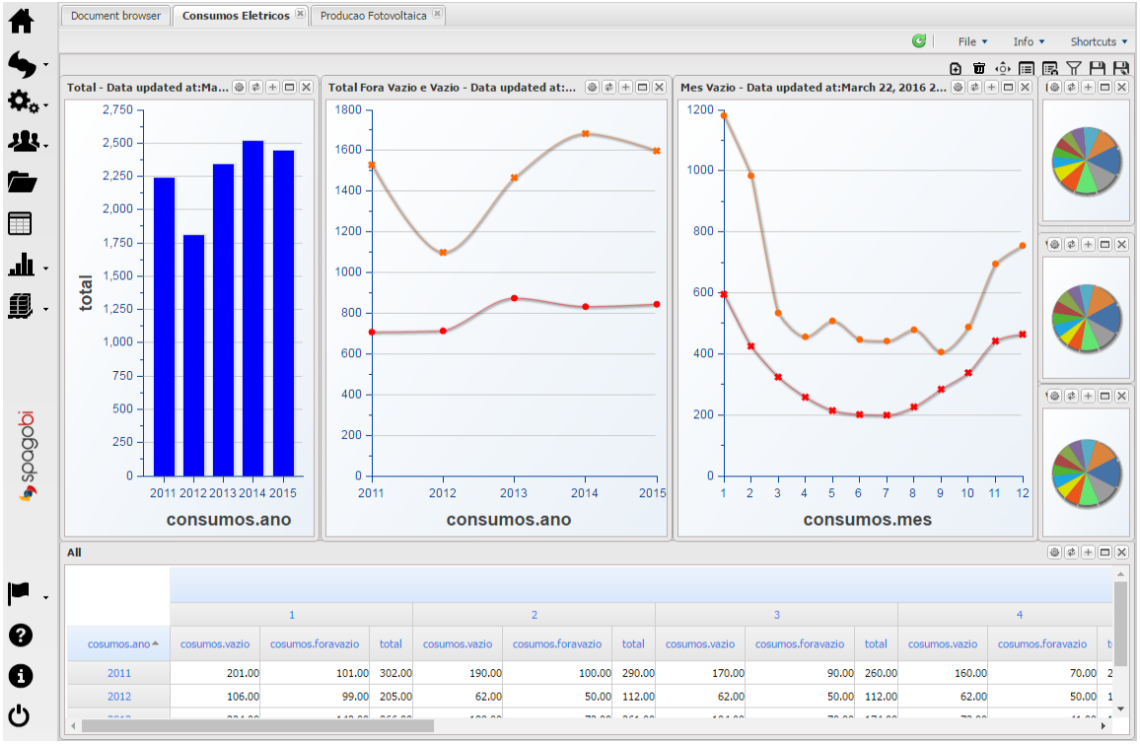


Figura 31 - Evolução consumos elétricos.

[6] http://gethue.com/

Na Figura 32 temos um outro *dashboard* que possibilita a análise comparativa entre a produção fotovoltaica e os consumos elétricos ao longo do tempo.

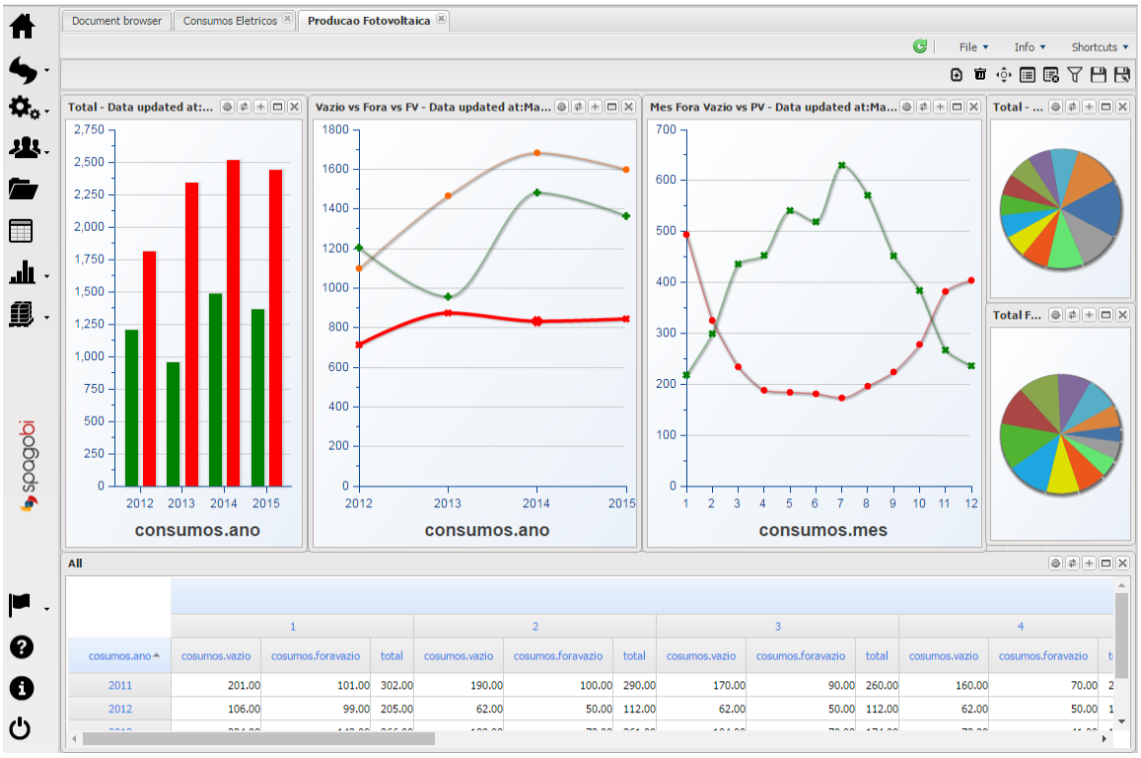


Figura 32 - Produção e consumos elétricos.

Testadas estas funcionalidades podemos concluir que é possível através do *SpagoBI* construir *dashboards* com base em tabelas do *Hive,* no entanto surgiu a necessidade de explorar as capacidades da plataforma no que diz respeito aos metadados. Como analisado no capítulo anterior, a arquitetura da *SpagoBI* disponibiliza um *plug-in* para o Eclipse com o nome *SpagoBI Meta* que possibilita, depois de definidos os metadados, o *upload* desta definição para o servidor e desta forma os dados ficam disponíveis para serem analisados nos diversos componentes analíticos. No entanto, era necessário que esta funcionalidade estivesse disponível diretamente no servidor no paradigma *as-a-Service*, para que, e de uma forma simples, os modelos de dados pudessem ser cridos e posteriormente proceder-se às suas análises de uma forma completamente livre. Identifica-se esta funcionalidade como sendo aquela que possibilitará o acesso à camada de armazenamento definido na arquitetura BASIS nomeadamente ao armazenamento operacional de *streaming* e histórico, e ao *analytics BDW*[7] *(Big Data Warehouse)*.

Foi seguida novamente a abordagem anteriormente descrita, de pesquisa na comunidade *SpagoBI* e através de *e-mail* com consultores da *SpagoBI*. Desta vez foi impossível

[7] No âmbito desta dissertação, o resultado da definição dos metadados que darão acesso aos dados será intitulada de dados da *smart city*.

encontrar uma solução passível de implementação em tempo útil, pois esta funcionalidade não está disponível, nem prevista, tanto ao nível da arquitetura do servidor *SpagoBI* como para futuro desenvolvimento, visto que esta é considerada uma funcionalidade de desenvolvimento.

A 08-04-2016 a *SpagoBI* lançou a versão 5.2 do produto, a qual foi imediatamente testada. No entanto, e por existirem diversos problemas, nomeadamente a instabilidade da plataforma, em 11-04-2016 foi disponibilizada uma nova *release* desta versão, sendo que esta foi instalada e testada com sucesso no exemplo descrito anteriormente.

Um dos serviços que importa disponibilizar será a análise de ficheiros que o utilizador possa ter na sua posse, como por exemplo, ficheiros pessoais que pretenda analisar. Posto isto, interessa testar a funcionalidade de envio de ficheiros diretamente para a plataforma para posterior análise, sendo que o servidor *SpagoBI* permite o envio de *datasets* no formato *csv* ou *xls* onde posteriormente podem ser feitas análises. A título de experimentação foi usado um ficheiro *xls*, como visível na Figura 33. De destacar que devem de ser definidos os elementos que são considerados atributos e métricas para uma posterior análise. Ainda neste ambiente é possível fazer uma pré-visualização dos dados em formato tabelar. Sobre este *dataset* podem ser agora feitas as análises de uma forma livre.

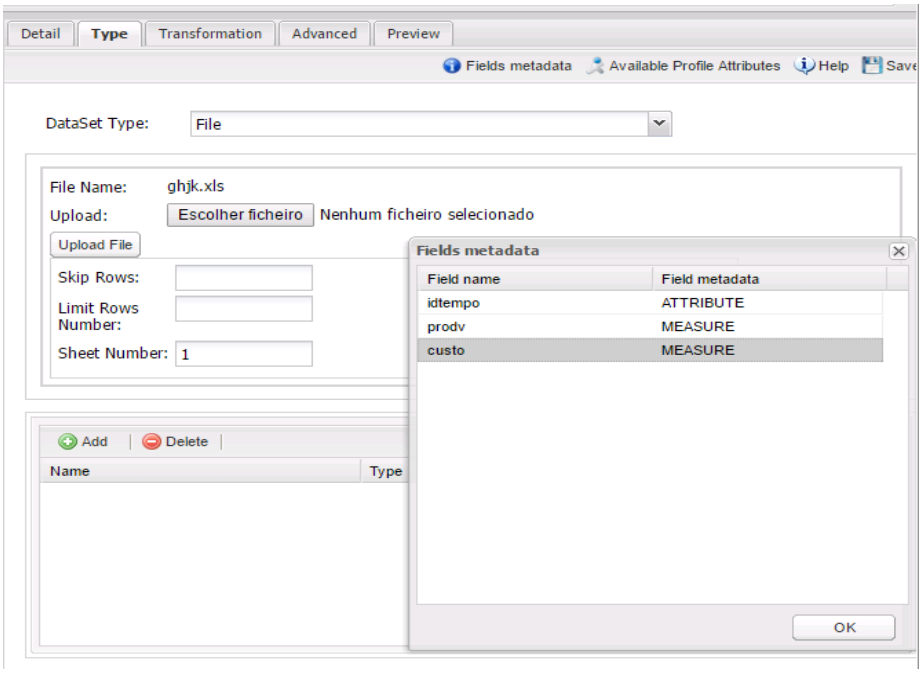


Figura 33 - *Upload* de ficheiro *xls* para plataforma *SpagoBI*.

A análise multidimensional ou outra equiparada fará parte da arquitetura que se pretende desenvolver, é necessário testar como a plataforma *SpagoBI* lida com esta temática. Foi então testada a funcionalidade *OLAP* assumindo que existem um *Data Warehouse* tradicional.

Para a criação de cubos *OLAP* é necessário utilizar o *SpagoBI Meta*. Como visível na Figura 34, foi necessário instalar este componente na nossa infraestrutura de teste para então podermos elaborar o modelo *OLAP*.

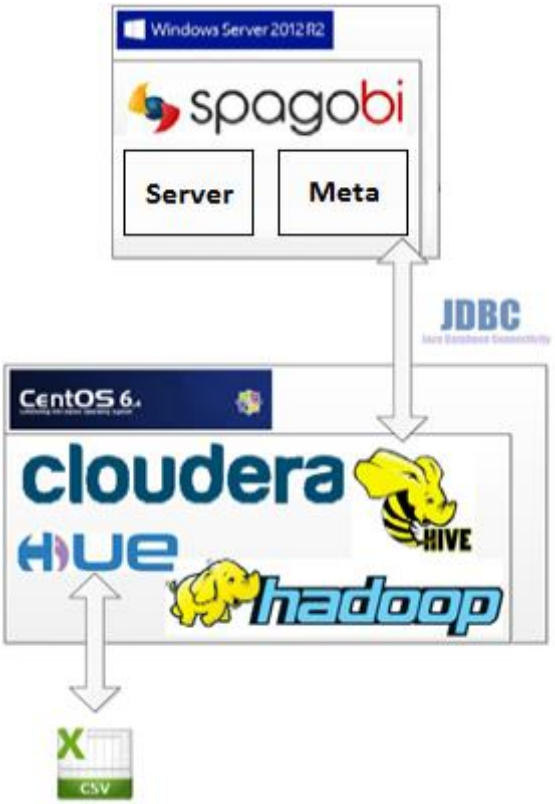


Figura 34 - Arquitetura base de teste da plataforma *SpagoBI Server* e *Meta*.

Foi também necessário utilizar os *drivers JDBC* para fazer a conexão do *SpagoBI Meta* ao *Cloudera*. Estando esta estabelecida é possível ter acesso às bases de dados assim como as tabelas que se encontram no *Hive*. Foi ainda necessário fazer a conexão ao *SpagoBI Server* para posteriormente ser feito o *upload* do acesso aos dados da *smart city*.

Depois de escolhidas as tabelas que serão usadas no cubo, é necessário definir quais serão as tabelas consideradas de dimensão e as que serão consideradas como tabelas de factos. Como é visível na Figura 35, a tabela Tempo foi definida como sendo a dimensão, as duas tabelas de factos, a de Produção e a de Consumos elétricos. É necessário em cada uma das tabelas definir as chaves e suas relações, os campos que serão considerados como atributos e os que serão considerados de métricas. Foi ainda definida a hierarquia da tabela Tempo. O acesso aos dados da *smart city* proposto encontra-se pronto a ser enviado para o servidor.

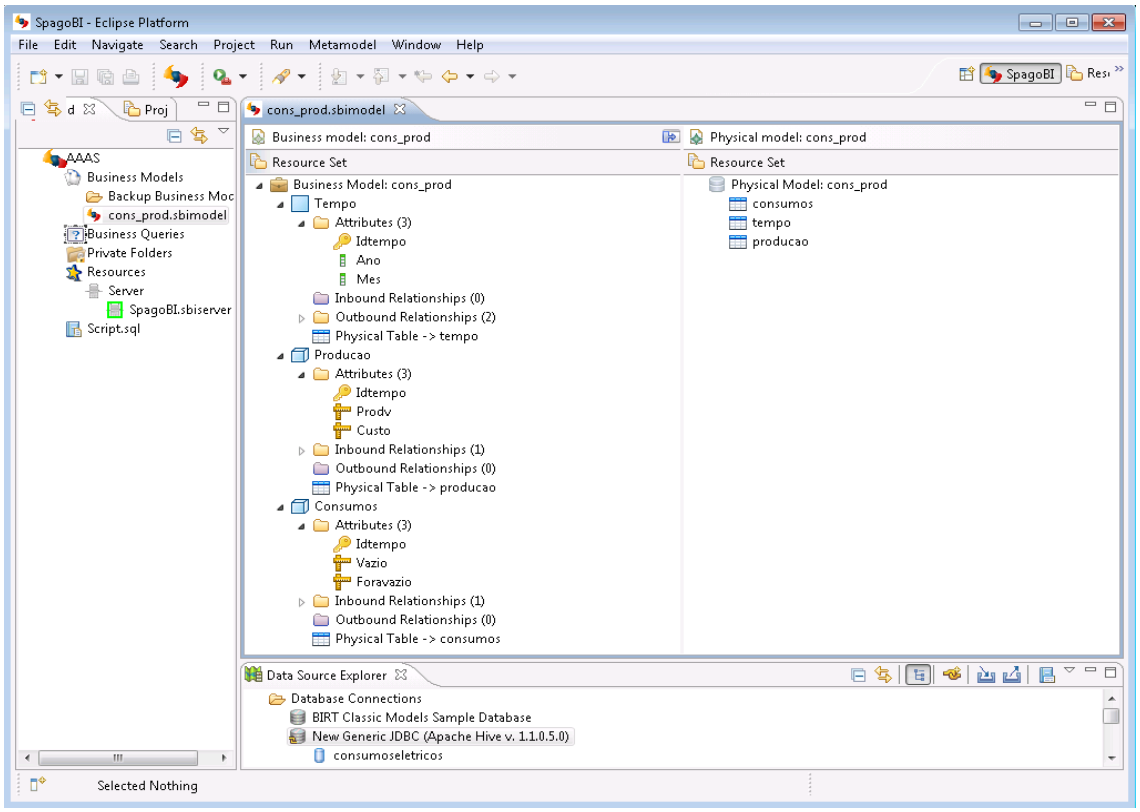


Figura 35 - Cubo *OLAP* em *SpagoBI Meta*.

Depois de enviado, como podemos visualizar na Figura 36, o acesso aos dados da *smart city,* fica disponível para ser explorado de uma forma livre, por exemplo através da funcionalidade *QbE (query by example)*.

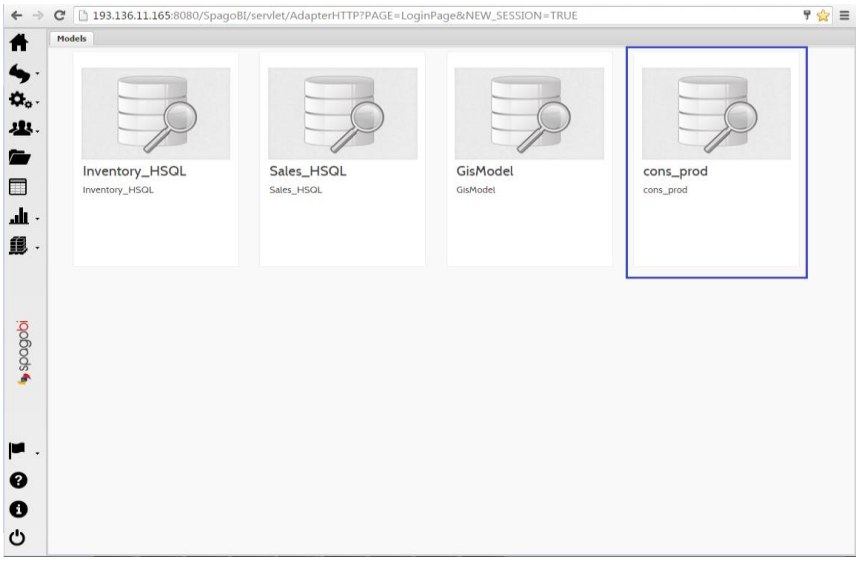


Figura 36 – Acesso aos dados da *smart city, OLAP* no servidor *SpagoBI*.

O *QbE* é uma funcionalidade que permite a análise de dados da *smart city* de uma forma livre e simples, arrastando os campos que pretendemos incluir na análise, a plataforma gera de uma forma transparente a *query* que representa a seleção dos campos sem que o

utilizador necessite de perceber o que quer que seja de programação. No caso do exemplo que estamos a utilizar, como visível na Figura 37, foram arrastados diversos campos que ditaram os dados para a nossa análise.

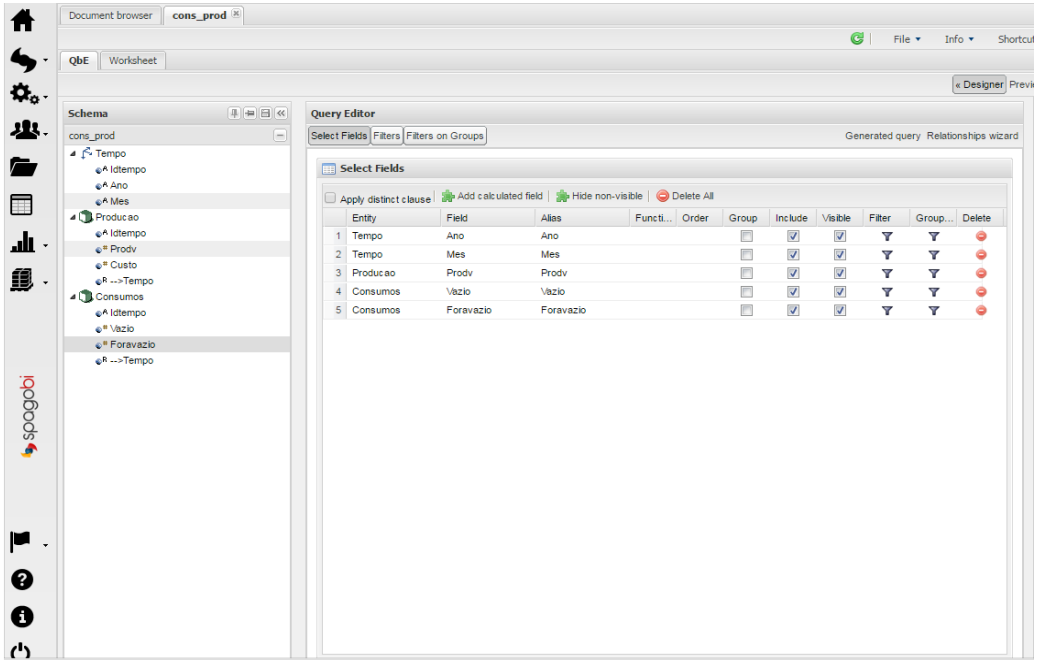


Figura 37 - *QbE* seleção para *query.*

Conseguimos pré-visualizar os dados, sendo possível exportar os mesmos em diversos formatos como por exemplo *xls*, *pdf* ou outros conforme visível na Figura 38.

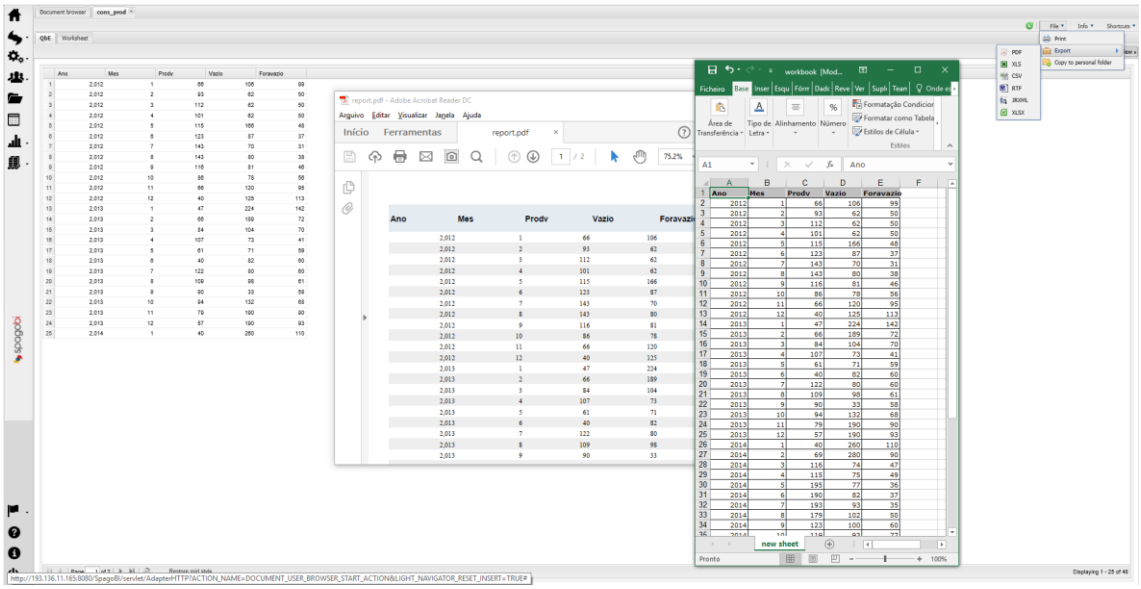


Figura 38 - Exportação de dados.

Feita esta seleção, como visível na Figura 39, é possível realizar a escolha dos atributos a serem analisados, assim como, o tipo de gráfico ou tabela que queremos que os mesmos sejam apresentados de uma forma visual. É ainda possível escolher quais os filtros a aplicar nos

dados bem como colocar diversos tipos de gráficos ou tabelas em diversas *sheets* de análise, onde cada uma delas pode ter um tipo de gráfico diferente. Depois de construído, este pode ser guardado para posterior consulta sendo que, a qualquer momento, pode ser alterado e ainda partilhado entre os diversos perfis de utilizadores existentes na plataforma.

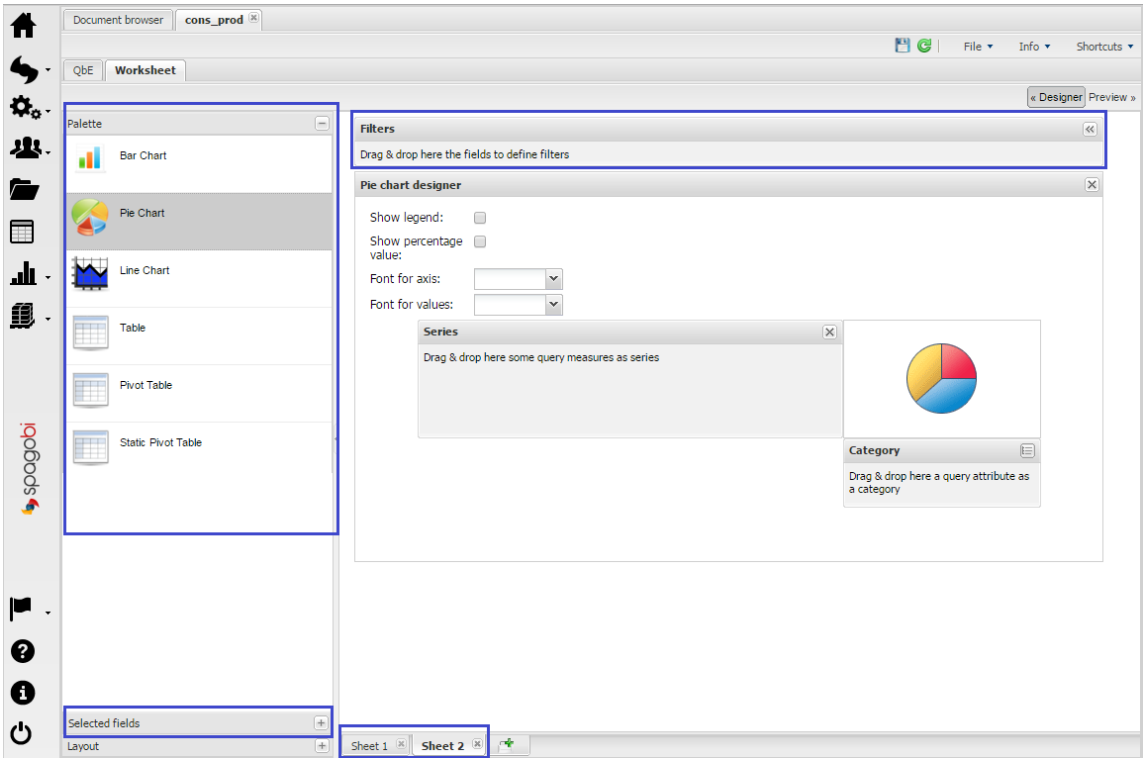


Figura 39 - Escolha de tipo de gráfico ou tabela.

A experimentação realizada, nas diversas vertentes analíticas, teve como objetivo perceber as funcionalidades e limitações do *SpagoBI* para o trabalho a realizar, de forma a melhor definir a arquitetura do sistema *DAaaS*.

4. DATA ANALYTICS-AS-A-SERVICE PARA SMART CITIES

Neste capítulo será apresentada a proposta de arquitetura definida para esta dissertação. Para tal adotou-se uma abordagem por camadas de abstração, sendo estas, divididas em camada conceptual e camada tecnológica, definindo ainda formalmente através de UML as principais funcionalidades. Esta proposta tem como objetivo definir uma arquitetura *DAaaS* que disponibilize o detalhe necessário para a criação de um ambiente analítico integrado, assente no conceito *as-a-Service*, com base na arquitetura BASIS e que supra as necessidades analíticas de uma *Smart City.* Como prova de conceito serão validados parte dos componentes propostos para a arquitetura BASIS, na disponibilização do serviço.

4.1. Proposta de detalhe analítico para a arquitetura BASIS

A plataforma *SpagoBI* descrita na secção 3.2 foi considerada como a que melhor se enquadra no âmbito desta dissertação. Esta foi validada, descrita a sua arquitetura e testada tecnologicamente através de pequenos exemplos descritos na secção 3.3. Nesta secção, desenvolve-se a proposta de arquitetura para um serviço *DAaaS* que supra as necessidades de uma *Smart City*.

Como observado em (Khan et al., 2013), os autores apresentam um estudo pertinente relativo à disponibilização dos dados num modelo *Analytics-as-a-Service.* No entanto, são pouco claros e não refletem na sua arquitetura quais os componentes necessários para que se possa oferecer este tipo de serviço. Ao nível conceptual referem algumas camadas, mas as mesmas não são descritas detalhadamente. Ao nível tecnológico, segundo os autores, a arquitetura é testada com sucesso (Khan et al., 2015). No entanto, os mesmos apenas descrevem a prova de conceito através de protótipo e testes de desempenho utilizando a arquitetura conceptual descrita em (Khan et al., 2013).

Em (Atos, 2013), foram delineadas linhas gerais para uma arquitetura deste tipo. No entanto, por um lado, são demasiado abrangentes, o que torna os conceitos pouco explícitos, por outro lado, os autores não referem especificamente os componentes tecnológicos para o desenvolvimento de uma plataforma deste tipo.

Em (Jara et al., 2014) e (Suakanto et al., 2013), a abordagem adotada está centrada nos dados gerados por sensores, pelo que os dados gerados por outros intervenientes não são

tidos em consideração. Apesar de se reconhecerem resultados satisfatórios por parte dos autores, as abordagens servem propósitos muito específicos e apenas cobrem parte dos objetivos de uma arquitetura capaz de lidar com as análises no paradigma *as-a-Service*.

Como observado em (Barga et al., 2012), estes desenvolvem uma proposta interessante na medida em que propõem um serviço baseado na *cloud* otimizado para a análise e para a aprendizagem automática. Ao mesmo tempo integram este serviço com ferramentas utilitárias como a folha de cálculo, sendo que é através de um *add-in* que o acesso ao serviço é disponibilizado. Exatamente por esta proposta ser baseada em ferramentas proprietárias, locais, e pela necessidade de instalação de um *add-in*, esta abordagem fica fora do paradigma *as-a-Service*.

Devido a estas conclusões, retiradas da análise das cinco arquiteturas aqui expostas, pode-se afirmar que a arquitetura aqui proposta não tem por base a alteração de nenhuma das soluções existentes, mas sim, o aprofundar e detalhar dos componentes apresentados na arquitetura BASIS. Qualquer um dos trabalhos apresentados se integrados na arquitetura BASIS não responderiam a alguns dos requisito do serviço que se pretende desenvolver, algumas delas por serem demasiado generalistas e não ter o detalhe tecnológico, outras por servirem propósitos específicos e outras por não estarem desenvolvidas no paradigma *as-a-Service*. No entanto, algumas das linhas gerais aqui discutidas são tidas em consideração no desenvolvimento de uma arquitetura que permita concretizar um *DAaaS* no contexto de *Smart Cities*. Esta proposta assenta no detalhe conceptual e tecnológico da arquitetura BASIS bem como na descrição formal em UML das principais funcionalidades do *DAaaS*.

4.1.1. Proposta Conceptual

A camada conceptual desta proposta de arquitetura visa encapsular os diversos componentes, nomeadamente os de processamento, análise, visualização, bem como os que servem de suporte para o funcionamento destes. O detalhe destes componentes tem como principal objetivo desambiguar tecnologicamente esta arquitetura, suprindo assim a necessidade encontrada na análise das arquiteturas aqui expostas e colmatando a necessidade apontada pelo autor da arquitetura BASIS. Como descrito no capítulo 3, em (Costa, 2015), é descrita a arquitetura BASIS. Nesta encontramos estes componentes descritos de uma forma superficial, apontando linhas gerais que servem de base a este desenvolvimento.

Na Figura 40, podemos visualizar os diversos componentes da arquitetura BASIS, integrada com os componentes analíticos propostos neste trabalho para o desenvolvimento do

serviço analítico que se pretende. São ainda destacados alguns componentes que já existem na arquitetura BASIS, que pela integração de novos componentes analíticos considera-se necessário aprofundar o seu detalhe, nomeadamente o *upload* de ficheiros, administração e modelos analíticos.

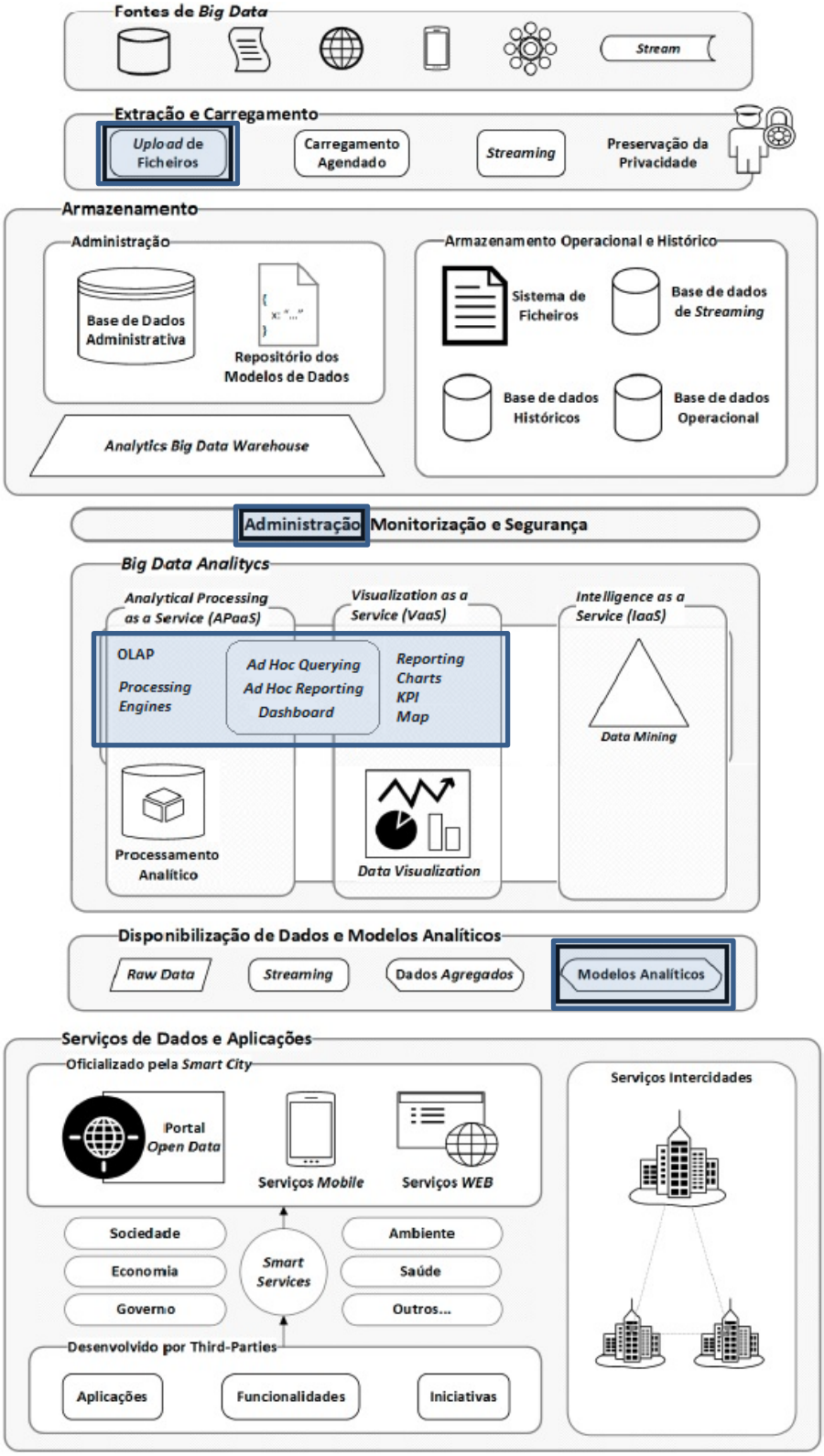


Figura 40 – Proposta de detalhe à camada conceptual da arquitetura BASIS.

O bloco *Big Data Analytics* é o principal componente desta arquitetura. É neste que os seus intervenientes podem retirar o valor dos dados. Dentro deste encontramos componentes de *Analitical Processing*, de *Data Visualization* e ainda outros que requerem os dois anteriores. Aqui é importante que os diversos componentes permitam, em estágios finais de análise, e sempre que os componentes deem origem a documentos ou tabelas de dados, a sua exportação em diversos formatos assim como a sua impressão.

Começando pelo componente *OLAP* que se encontra dentro do *Analytical Processing as a Service,* este tem como principal objetivo fazer o processamento *OLAP*, providenciando a capacidade de manipular e analisar um grande volume de dados sob múltiplas perspetivas. Como estamos num contexto de *Big Data,* as abordagens convencionais *OLAP* simplesmente não funcionam pelo que se aconselha a utilização de novas técnicas para se lidar com este novo paradigma, como a solução apresentada em (Santos & Costa, 2016), ou outra, que se revele capaz de suprir a necessidade de análise de grandes volumes de dados sob múltiplas perspetivas. Ainda dentro desta camada podemos encontrar os motores analíticos que têm como objetivo processar os dados requeridos pelos componentes *Ad Hoc Querying*, *Ad Hoc Reporting* e *Dashboard*.

Em relação ao *Data Visualization as a Service,* encontramos as diversas formas de visualização de dados, onde se propõe a existência dos seguintes componentes:

- *Reporting* – Tem como objetivo providenciar a capacidade de geração de relatórios quer estáticos, quer dinâmicos. Neste último, a inserção de parâmetros por parte do utilizador deve permitir o retorno do resultado da pesquisa. Deve ainda ser capaz de exportar os relatórios em diversos formatos;
- *Charts* – A par do anterior, este deve permitir visualizar e exportar gráficos de diversas formas e diferentes formatos. Deve permitir a interação com o utilizador de uma forma dinâmica através da navegação interativa dos mesmos;
- *KPI* – Deve permitir a visualização e definição de diversos indicadores;
- *Map* – Este componente, dada a sua especificidade, requer particular atenção visto ser necessário a integração de mapas. Os dados serão integrados com mapas, que permitirão sua visualização neste contexto.

Sobre os componentes que, dada a sua natureza, necessitam por um lado do processamento e por outro da sua visualização destacam-se os seguintes:

- *Ad Hoc Querying* – Deve ser possível, de uma forma *ad hoc* e transparente, o simples arrasto dos atributos ou métricas por forma a gerar um conjunto de dados específico. Estes podem ser analisados de uma forma interativa em gráficos e/ou relatórios, podendo ser guardados para posterior consulta e/ou partilha;
- *Ad Hoc Reporting* – Ligado diretamente ao *Ad Hoc Querying*, uma vez que providencia um conjunto de dados específico, este pode ser trabalhado em relatórios que podem ser guardados para posterior consulta. Este deve providenciar a criação de relatórios simples utilizando as diferentes fontes de dados disponíveis;
- *Dashboard* – Aqui deve de ser possível a escolha de dados provenientes dos diversos *datasets,* e sobre estes criar diversos componentes visuais, sejam eles gráficos de diversos tipos, tabelas simples ou com agregações. Sempre que exista relação direta entre os dados distribuídos pelos diversos gráficos e tabelas, o *dashboard* deve agir dinamicamente em todos eles.

O *upload* de ficheiros é um componente que está definido na arquitetura BASIS como um dos que contribuem para o aumento do volume e da variedade dos dados disponíveis. Não obstante desta premissa, o simples *upload* deste tipo de ficheiros para a plataforma não deve ter o único intuito de alimentar a plataforma, mas também ser possível a sua análise direta sem que exista a necessidade de passar pelos processos integradores nas bases de dados da plataforma. Estes ficheiros, e como aconselhado em (Costa, 2015), devem ficar guardados no repositório *HDFS* da plataforma. Como exemplo, um cidadão pode fazer *upload* do seu ficheiro de Excel que costuma usar para analisar os seus consumos elétricos ao longo do tempo e proceder à sua análise direta nos componentes analíticos da plataforma, criando gráficos e/ou relatórios que lhe permitam retirar valor dos dados.

No que diz respeito aos dados da *smart city* é necessário que estejam incluídas capacidades de criar acesso a esses dados disponíveis por forma a permitir a análise dos mesmos.

É necessário ter especial atenção aos acessos dados, aos dados da *smart city* e aos ficheiros que os diversos intervenientes colocam na plataforma, sob pena de tornar a mesma ingerível. Qualquer que seja o interveniente, deve ser elucidado que os ficheiros poderão ter características históricas, que devem de ser respeitadas. Estes ficheiros devem ter um caris temporal reduzido, pelo que deve ser dada a possibilidade de escolha ao interveniente, se este

deseja integrar os seus ficheiros nas bases de dados existentes, ou se serão apagados da plataforma após um espaço de tempo a definir. Cabe a cada iniciativa definir estes parâmetros para que a plataforma não se torne ingerível. Deve ser disponibilizada uma área de gestão (Administração), onde é dada a possibilidade aos utilizadores de apagar os seus ficheiros. A par desta cabe ao gestor da plataforma uma área reservada onde terá acesso a todos os ficheiros e aos dados da *smart city,* que se encontram na plataforma.

Depois de descrito conceptualmente o detalhe analítico desta proposta, através da descrição dos componentes, e das suas principais relações, seguidamente será apresentada a proposta tecnológica da arquitetura.

4.1.2. Proposta Tecnológica

Depois da arquitetura estar definida conceptualmente é considerado importante instanciar a mesma com as tecnologias necessárias para que esta possa ser posta em prática. Não obstante de poderem ser utilizadas outras tecnologias, desde que cumpram os princípios conceptuais definidos anteriormente, as tecnologias usadas têm como princípio fundamental o uso de licença *open source.* Serve de base desta proposta os componentes *SpagoBI* e seus motores, sendo que alguns destes componentes não estão desenvolvidos no paradigma *as-a-Service,* nomeadamente o *SpagoBI Meta e SpagoBI Studio*, mas como esta plataforma é totalmente *open source*, tanto para utilização, como para desenvolvimento, poderá servir de base para o desenvolvimento dos mesmos no paradigma *as-a-Service*. Conforme verificado anteriormente, os autores (Khan et al., 2013), (Khan et al., 2015), (Atos, 2013), (Jara et al., 2014), (Suakanto et al., 2013) e (Barga et al., 2012), apontam algumas tecnologias, no entanto muitas vezes não as instanciam na sua arquitetura.

Tal como demonstra a Figura 41, a camada tecnológica contempla os componentes principais de processamento e de visualização analítica, assim como os componentes que envolvem toda a temática e troca de informação entre componentes, nomeadamente os que estão definidos em (Costa, 2015). Para além destes, esta tem como finalidade acrescentar à sua arquitetura especificidades ao nível das tecnologias que sustentam a proposta aqui apresentada.

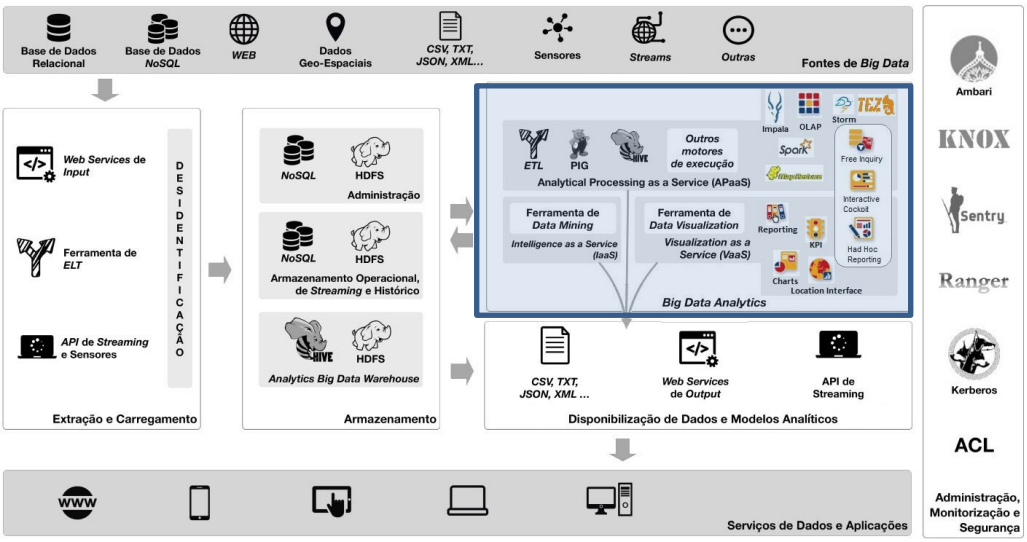


Figura 41 – Proposta de detalhe da camada tecnológica da arquitetura BASIS.

Na Figura 42, em destaque, podemos observar o detalhe dos componentes propostos.

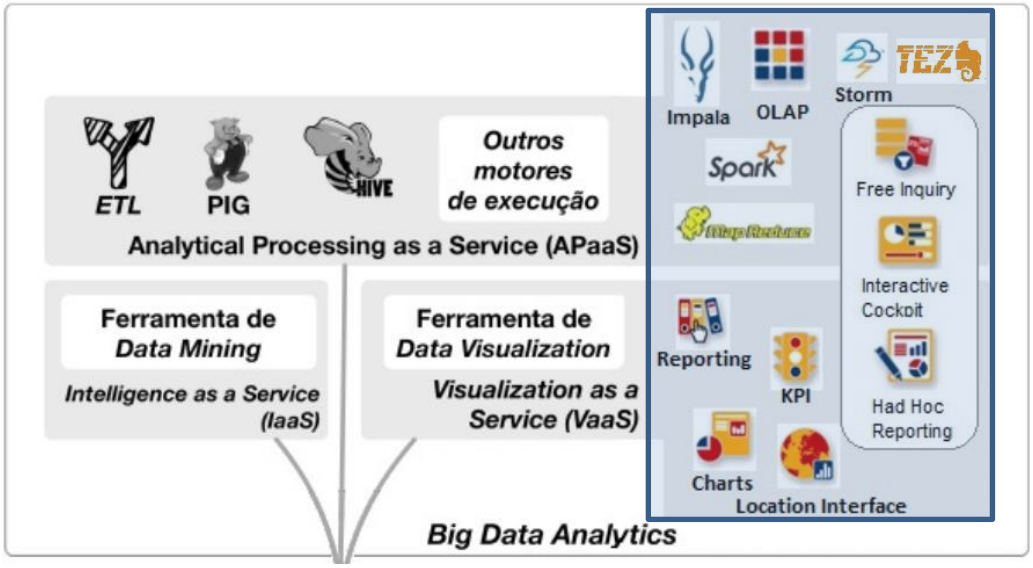


Figura 42 – Detalhe de componentes de *Big Data Analytics.*

O *Data Analytics* está fortemente presente no *SpagoBI*, sendo que esta plataforma, como analisado anteriormente, contém um vasto leque de motores analíticos disponíveis no paradigma *as-a-Service*. No entanto, esta necessita ainda de uma mudança de paradigma bastante acentuada, se o ambiente de aplicação assim o exigir, nomeadamente a migração da sua base de dados operacional de *HyperSQL* para o paradigma *Big Data.*

Dentro da subcamada *Analytical Processing as a Service* encontramos:

- *Analytical Processing as a Service:* Na plataforma *SpagoBI* encontramos o processamento *OLAP* como sendo a sua principal característica. No entanto, será aconselhado o desenvolvimento de suporte, na plataforma *SpagoBI Server*, para os *drivers* que tenham por base os motores mais ágeis como o

Impala[8] e/ou *Spark* e/ou *Storm* por forma a agilizar o processamento e por consequência a visualização de grandes volumes de dados. Visto que o *SpagoBI* tem suporte *HIve*, a utilização do motor *Tez*[9] poderá ser uma solução a ponderar visto que este promete melhor desempenho que o *MapReduce*. A criação de cubos *OLAP*, utilizando tabelas criadas diretamente no *Hive,* não será aconselhado em contexto *Big Data* na medida em que tarefas de *MapReduce* que utilizem operadores como o *join,* tornam o seu processamento computacionalmente custoso (Huang, Zhang, Buyya, Chen, & Wu, 2016). No entanto, a viabilidade desta abordagem poderá tornar-se possível se forem utilizas tabelas pré-computadas e que respondam a uma ou mais questões específicas, onde a quantidade de dados é consideravelmente menor;

Data Visualization as a Service: Em relação ao *Reporting,* o *SpagoBI* tem dois motores, o *BIRT*ReportEngine e o JasperReportEngine, que tiram partido das características do *BIRT* e do *Jaspersoft*, sendo que neste componente apenas está disponível a elaboração de relatórios simples já que, relatórios complexos são assegurados pelo componente *SpagoBI Studio*. Os gráficos presentes estão assegurados pelo motor *ChartEngine* que contém diversos tipos de gráficos. A geração de *KPI's* é também assegurada pela plataforma, bem como a geração de relatórios e visualização dos mesmos. O *Location Interface* está assegurado por dois motores, o *GeoEngine* para visualização de dados em mapas e o *GeoReportEngine* para a criação de relatórios em mapas.

Existem ainda componentes que, por um lado, necessitam de processamento analítico e por outro a sua visualização. Aqui podemos encontrar:

- *Ad Hoc Querying:* Este é assegurado pelo motor *QbeEngine* providenciando a capacidade de geração de *queries* de uma forma flexível e transparente aos dados da *smart city* previamente concebidos. O resultado deste é um *dataset* que pode posteriormente ser trabalhado e visualizado em relatórios e gráficos;

[8] http://impala.io/
[9] http://tez.apache.org/

- *Interactive Dashboard:* Assegurado pelo *CockpitEngine,* permite a escolha de diversos gráficos e tabelas para análise de dados. Um mesmo *dashboard* pode conter mais que uma origem de dados. No entanto não permite, num mesmo gráfico, mais que uma origem de dados. Aqui seria aconselhado o desenvolvimento da funcionalidade de cruzar mais que uma fonte de dados num mesmo gráfico, de uma forma completamente livre;
- *Ad Hoc Reporting:* Podem ser feitos relatórios simples baseados nos dados disponíveis. Esta funcionalidade está ainda ligada com as duas anteriores pois estas permitem guardar determinado conjunto de dados que podem ser disponibilizados em forma de relatório.

Depois de instanciados os componentes tecnológicos necessários à concretização desta arquitetura ficou claro que existe a necessidade para a implementação completa desta plataforma, o ajuste e o desenvolvimento de alguns dos componentes. No entanto, na subsecção 4.3 será feita a prova de conceito que demonstrará ser possível implementar sem qualquer desenvolvimento parte da arquitetura proposta. A decisão do desenvolvimento dos restantes componentes fica fora do âmbito desta dissertação. No entanto, ficam apontadas as linhas gerais necessárias para o seu futuro desenvolvimento.

Seguidamente é feita a proposta de funcionalidades *DAaaS*.

4.2. Proposta de funcionalidades de *DAaaS*

Esta secção tem como objetivo detalhar as principais funcionalidades propostas para o sistema *DAaaS*. Para este detalhe foi desenvolvido em UML os casos de uso de primeiro e segundo nível bem como as suas descrições. Dá-se principal destaque às funcionalidades analíticas da proposta, pelo que para estas foram feitos casos de uso de terceiro nível, visto que este é o foco desta dissertação. Quanto às funcionalidades de gestão, apesar de necessárias não se considera relevante a sua descrição detalhada em casos de uso de mais baixo nível, pois estas são consideradas de suporte, não tendo propriamente que ver com o serviço analítico proposto, no entanto são descritas as linhas gerais à concretização das mesmas. Os casos de uso e as funcionalidades propostas encontram-se em inglês[10] para que seja mais fácil a sua instanciação nas tecnologias propostas.

[10] Nesta secção visto que as funcionalidades estarão descritas em inglês quando referido *Smart City Data* estamos a referir aos acessos a dados da *Smart City*.

Este primeiro nível denominado de *DAaaS* está dividido em cinco casos de uso, como se pode verificar na Figura 43. A descrição é apresenta abaixo na Tabela 9 e Tabela 10.

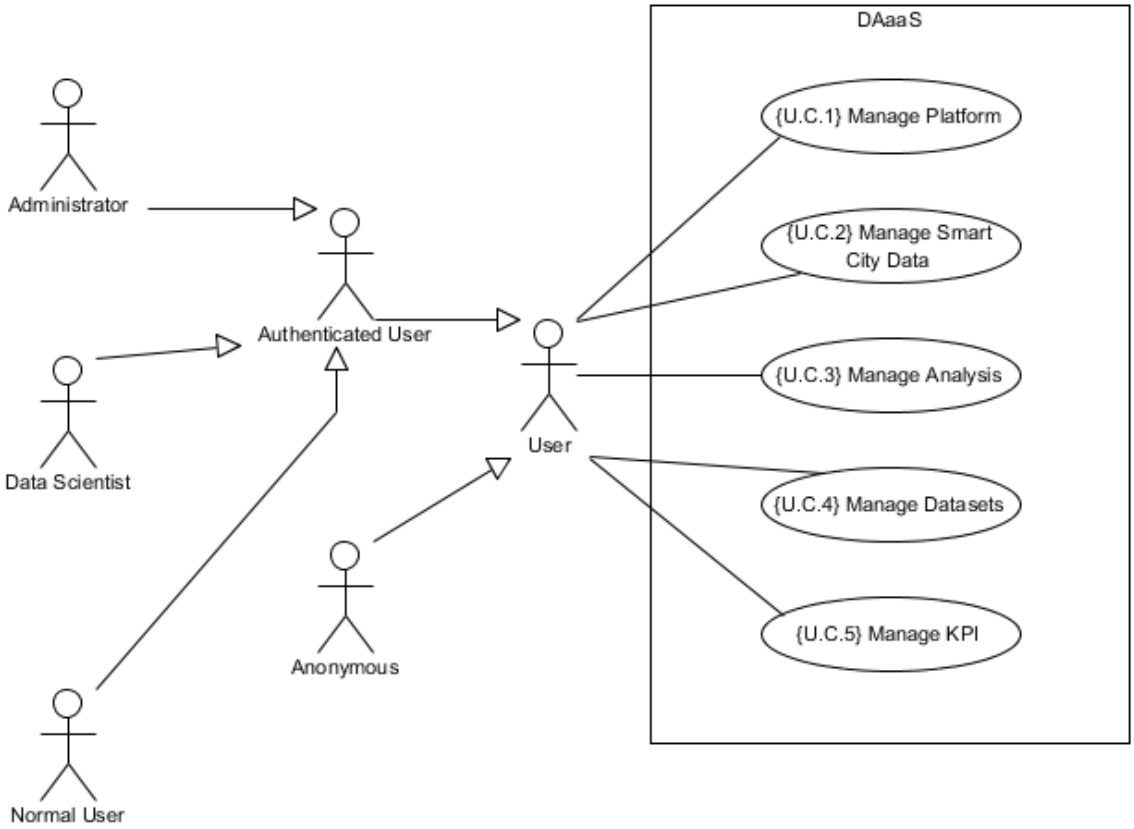


Figura 43 - Proposta de funcionalidades *DAaaS*.

Atores/Intervenientes	
Administrator	Ator responsável pela administração do sistema, que tem permissões para executar todas as funcionalidades.
Data Scientist	Ator que tem permissões para executar funcionalidades avançadas de gestão de modelos, gestão de análises, gestão de dados e gestão de *KPI*. Podendo ver e alterar o que for inserido na plataforma pelo *User*. Este ator pode efetuar operações sobre todos os *datasets* criados.
Normal User	Ator que tem permissões para executar funcionalidades de gestão de modelos, gestão de análises, gestão de dados e gestão de *KPI*. Neste ator estão incluídos os cidadãos, empresas ou outro interveniente que pretenda usar o serviço. Este ator apenas pode efetuar operações sobre os *datasets* criados por si ou partilhados.
Authenticated User	Ator registado no sistema, com acesso a um conjunto de serviços não disponíveis para utilizadores não autenticados *(Anonymous).*

Tabela 9 - Descrição dos atores/intervenientes.

Atores/Intervenientes	
Anonymous	Ator não registado no sistema, com acesso limitado apenas a serviços de visualização e análise de dados disponíveis e partilhados, sem poder alterar ou criar.
User	Utilizador do sistema, independentemente de ser utilizador autenticado ou não.

Tabela 9 - Descrição dos atores/intervenientes.

DAaaS	
{U.C.1} Manage Platform	Permite gerir a plataforma.
{U.C.2} Manage Smart City Data	Permite gerir os modelos e os dados da *smart city*.
{U.C.3} Manage Analysis	Permite gerir as análises.
{U.C.4} Manage Datasets	Permite gerir *datasets*.
{U.C.5} Manage KPI	Permite gerir *KPI*.

Tabela 10 - Especificação do caso de uso *DAaaS*.

Os casos de uso que detalham as funcionalidades propostas encontram-se no anexo A.

4.3. Implementação e demonstração de parte da arquitetura *DAaaS*

Para executar a instanciação da arquitetura definida na secção 4.1, foi integrado no protótipo do "SusCity" o *SpagoBI Server*. O "SusCity" apresenta-se como um projeto MIT Portugal financiado pela FCT, Fundação para a Ciência e Tecnologia, Ministério da Educação e Ciência, EDP Distribuição, ADENE, R&D Nester; Novabase e ITds.

O projeto "SusCity" "é focado no desenvolvimento e integração de novas ferramentas e serviços para aumentar a eficiência dos recursos urbanos, com impactos ambientais mínimos, contribuindo simultaneamente para promover o desenvolvimento económico e preservar os níveis reais de confiabilidade." ("SusCity," 2016). Neste protótipo de *DAaaS* foram integradas as funcionalidades dos componentes *SpagoBI, Interative Dashboard*; *Free Inquiry; Upload* de ficheiros. Esta implementação tem não só o propósito de servir de prova de conceito, como demonstrar como o cidadão pode usar parte do serviço analítico proposto. Como fonte de dados serão utilizadas duas origens, uma através do *upload* direto para a plataforma de um ficheiro *xls* e sua análise, a segunda será a utilização de uma tabela *Hive* simulando a utilização de um *Data Warehouse* em contexto de *Big Data.*

Para a integração de parte das funcionalidades propostas na secção 4.2, no portal "SusCity", foi utilizado como base o *SpaboBI Server,* onde se encontram grande parte das funcionalidades propostas. Para ser possível a utilização dos *SpagoBI Server,* no ambiente "SusCity", foi necessário efetuar alguns desenvolvimentos. Primeiramente foi desenvolvido um componente que permite no ambiente "SusCity", o acesso ao *SpagoBI Server,* mantendo inalterado o ambiente do "SusCity". Foi verificado que nativamente, para se aceder aos componentes analíticos no *SpagoBI Server* é necessário efetuar *login,* com um utilizador e uma *password* registada no sistema. Pela necessidade de o serviço estar aberto a qualquer utilizador que aceda ao portar mesmo que não esteja registado, foi necessário desenvolver uma abstração que permitisse entrar no serviço analítico sem existir a necessidade de efetuar *login*. Esta abstração foi desenvolvida e posta em prática, sendo que foi necessário criar um utilizador "anónimo" com parametrizações específicas. Esta abstração não invalida que um utilizador registado no sistema possa fazer *login* com as suas credenciais e assim aceder ao seu ambiente analítico reservado. Para estes desenvolvimentos foi utilizado o *IDE NetBeans 8.1 JavaEE* a correr com servidor *Apache Tomcat 8.0.27*. Os motores utilizados foram o *QbeEngine*, *CockpitEngine* e o *ReportEngine* juntamente com a funcionalidade do *SpagoBI Server* de *upload* de ficheiros. Para além dos desenvolvimentos descritos, para que esta abordagem fosse possível foi necessário fazer parametrizações no *SpagoBI Server*, nomeadamente a criação de utilizadores de teste, definição de ambientes, criação de categorias e criação de perfis. Algumas das funcionalidades descritas na subsecção 4.2, ficaram disponíveis para serem usadas. Depois desta implementação o serviço analítico ficou temporariamente disponível em (http://94.61.195.214:8084/analytic.html)[11].

Para a demonstração do *Data Warehouse* em contexto de *Big Data* irá ser utilizado um conjunto de dados que estão distribuídos por vinte e sete ficheiros no formato *csv* retirados de (RIBA-BTS, 2015), repositório que contêm dados sobre os vários voos dos aeroportos dos EUA, e que podem ser utilizados sem qualquer restrição. Dos vinte e sete ficheiros que foram utilizados, vinte e dois correspondem a dados referentes aos voos e os restantes a informação auxiliar. O conjunto destes corresponde a um agrupamento de informação recolhida no período desde 1987 a 2008 num total de 123534969 registos. Não foram utilizados para visualização os dados referentes ao ano de 1987, devido à inexistência de registos para os primeiros meses do ano.

[11] *Link* apenas disponível por pedido ao autor desta dissertação.

Tendo em conta o que foi descrito em (Santos & Costa, 2016), foi estudada uma nova abordagem sobre o conceito de *Data Warehouse* e a sua implementação em *Hive*. Neste estudo foi apresentado um conjunto de cinco regras que permitem transformar um modelo multidimensional num modelo *Big Data*, ou seja colocar um *Data Warehouse* tradicional em um *Data Warehouse* em *Big Data*.

Após ter sido estudada a abordagem proposta em (Santos & Costa, 2016), esta foi posta em prática seguindo as regras para a definição de uma tabela em *Hive*. A construção desta tabela *Hive* foi feita com a junção das tabelas *Dim_Carrier, Dim_Local, Dim_Time_Calendar, Dim_Time_Hour*, com a tabela de factos *Fact_Delays*, que deu origem à tabela aqui usada para demonstração. Esta tabela foi colocada na plataforma *Cloudera* com o nome Voos, em que a sua concretização está detalhada em (Martinho, 2016).

Na Tabela 11 estão descritos os atributos do *BDW*, Voos, correspondente aos atrasos dos voos, usado para a demonstração do serviço analítico.

Tabela 11 - Descrição dos atributos do *BDW* Voos.

Variável	Descrição
Year	Ano em que o voo ocorreu
Quarter	Quarto do ano em que o voo ocorreu
Month	Mês
DayOfWeek	Dia da semana
Carrier_Name	Nome da companhia aérea
Dest_Airport_Name	Nome do aeroporto de chegada
Dest_City_Name	Nome da cidade do aeroporto de chegada
Dest_State_Name	Nome do estado do aeroporto de chegada
Origin_Airport_Name	Nome do aeroporto de partida
Origin_City_Name	Nome da cidade do aeroporto de partida
Origin_State_Name	Nome do estado do aeroporto de partida
Interval_hour_arr	Intervalo de horas de chegada
Interval_hour_dep	Intervalo de horas de partida
Number_of_minutes_depdelay	Atraso, em minutos, de partida
Number_of_minutes_arrdelay	Atraso, em minutos, de chegada
Number_of_minutes_total_delay	Total de atraso, em minutos
Number_of_minutes_delay_company_reasons	Atraso, em minutos, devido a razões da companhia
Number_of_minutes_delay_metereological	Atraso, em minutos, por razões meteorológicas
Number_of_minutes_delay_congestion_reasons	Atraso, em minutos, por razões de congestionamento
Number_of_minutes_delay_security_reasons	Atraso, em minutos, por razões de segurança
Number_of_minutes_delay_reasons_other_aircraft_delay	Atraso, em minutos, por razões de atraso de outro voo
Flag_delay_15_in_minutes	Flag que indica se o voo chegou atrasado mais do que 15 minutos
Flag_voo_accomplished	Flag que indica se o voo foi realizado
Arr_Delay_15	Flag que indica se o voo chegou atrasado mais do que 15 minutos

Na Figura 44, estão destacados os componentes da arquitetura que serão usados e testados.

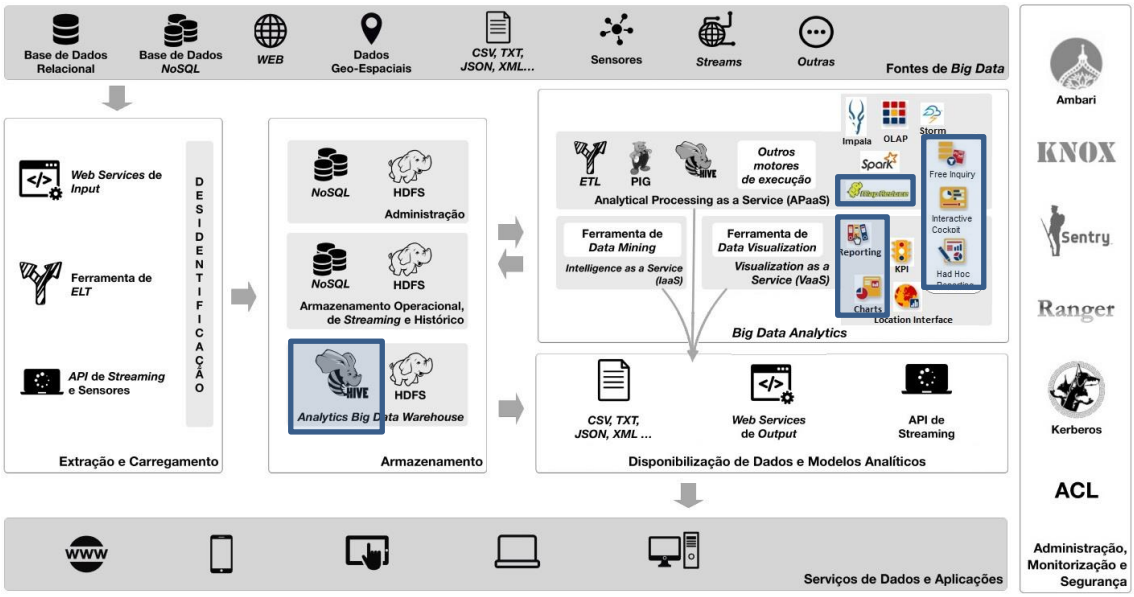


Figura 44 – Destaque de componentes usados.

Para a integração com as funcionalidades propostas, no protótipo "SusCity", foi criada uma nova categoria com a designação *My SusCity Data* destacado na Figura 45. Esta categoria de *Data Layer* dá acesso ao ambiente analítico criado com base nos componentes destacados na Figura 44.

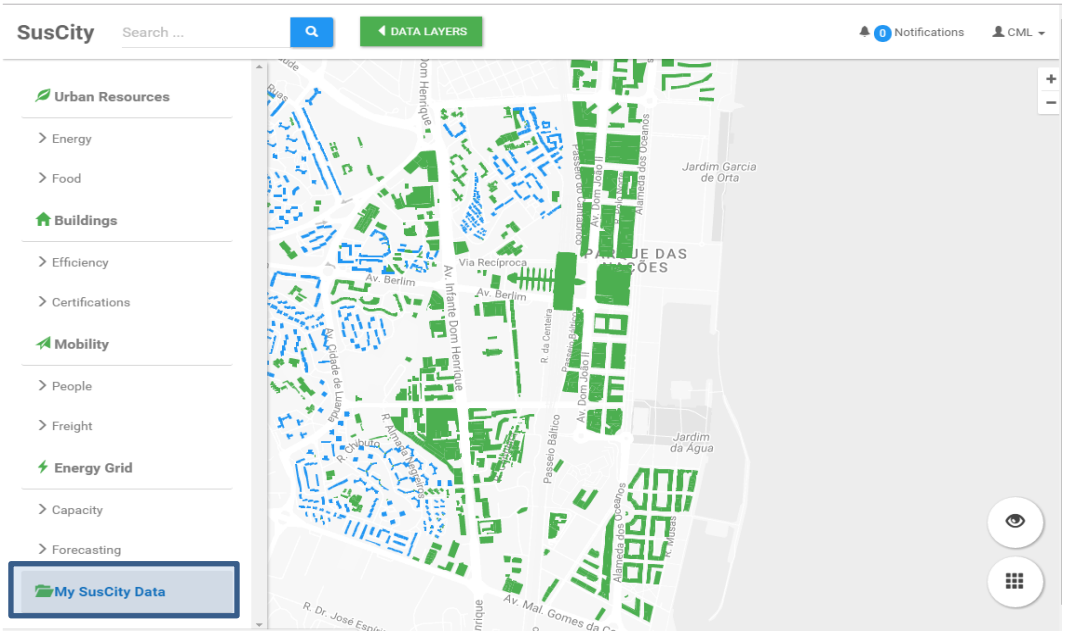


Figura 45 - *My SusCity Data.*

Na Figura 46, Figura 47, Figura 48 e Figura 49 estão visíveis parte das funcionalidades apontadas na subsecção 4.2. De seguida é dado o detalhe de cada uma das figuras.

Na Figura 46 é visível o ambiente inicial do *My SusCity Data*, destacado temos dois grupos de funcionalidades disponíveis, *My Analysis* (*{U.C.3} Manage Analysis*) e *My Data* (*{U.C.4} Manage Datasets*). Sucintamente, na funcionalidade intitulada de *My Analysis* existe a possibilidade de criar relatórios e *dashboards* com base nas origens de dados disponíveis, sendo ainda possível aceder a relatórios e a *dashboards* já criados. Em relação à funcionalidade *My Data*, está disponível, o *upload* de ficheiros, *xls* ou *csv*, por forma a guardar *datasets* para análise. Disponibiliza ainda o acesso a *datasets* já existentes e aos acessos a dados da *Smart City* já criados, assim como a aplicação da funcionalidade de *ad hoc queries* a estes.

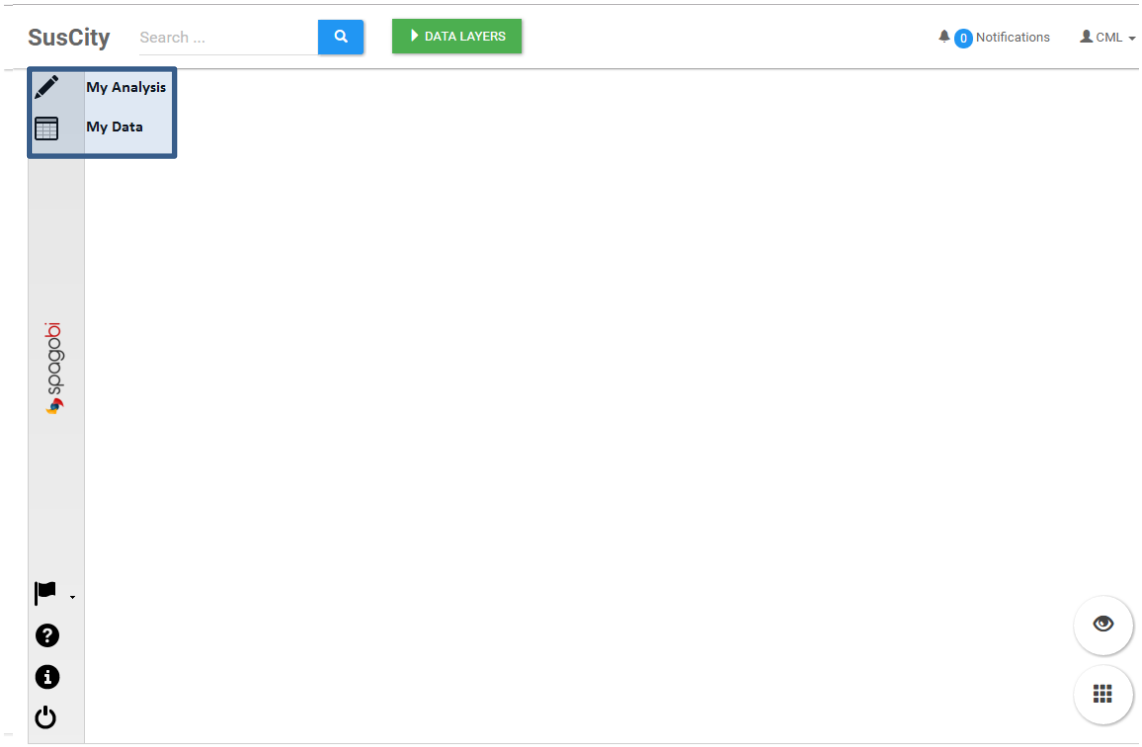


Figura 46 – Grupos de funcionalidades disponíveis no protótipo *DAaaS*.

Com visível na Figura 47, destacadas, estão as funcionalidades disponíveis de *My Data.* O separador visível com *datasets* corresponde ao (*{U.C.4} Manage DataSets*) e *models* ao (*{U.C.2} Manage Smart City Data*). A funcionalidade destacada, e identificada com o número um, corresponde à partilha e à retirada de partilha de um *dataset* (*{U.C.4.3} Share Datasets*). Este ícone quando visível representa esta funcionalidade mesmo que associado a outra visualização, como por exemplo na Figura 48 (*{U.C.3.1.3} Share Reports*). Destacada com o número dois, descrevendo da esquerda para a direita os ícones, é possível ver e alterar os detalhes do *dataset* como métricas e atributos (*{U.C.4.5} Update Datasets*), criar gráficos e tabelas, criar *queries ad hoc* (*{U.C.4.2} Use Datasets*) e apagar o *dataset* (*{U.C.4.4} Delete Datasets*).

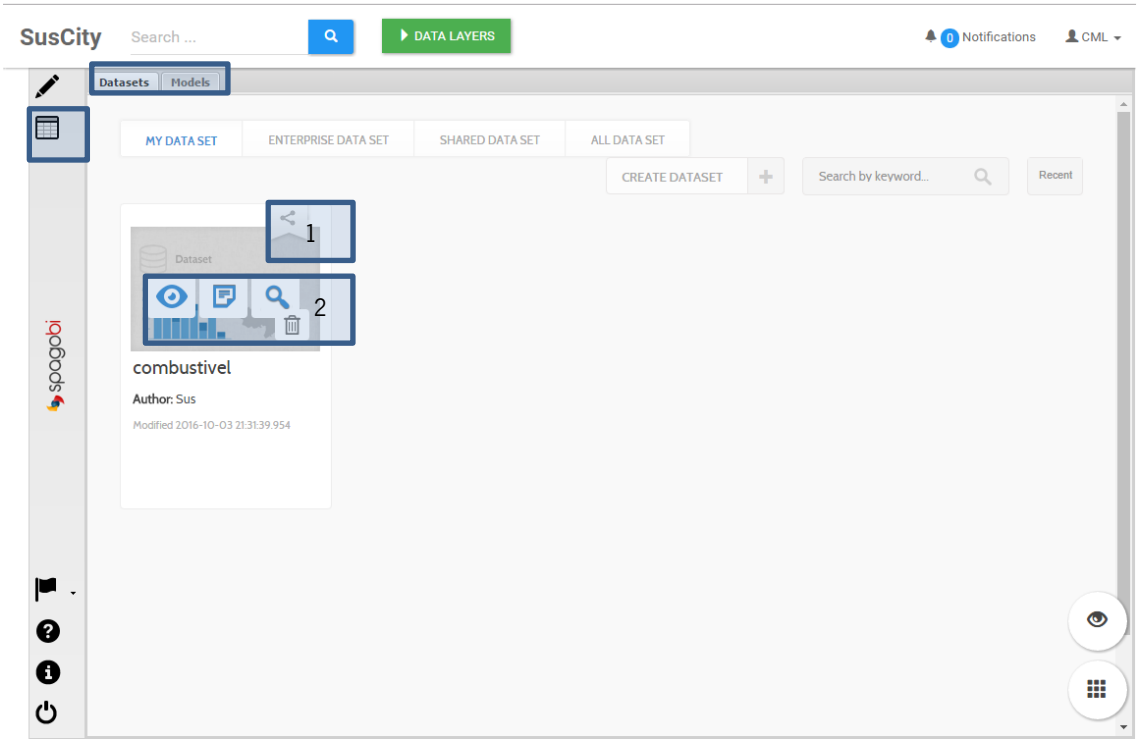


Figura 47 - Funcionalidades de *My Data.*

Como visível na Figura 48, estão destacadas as funcionalidades de *My Analysis*, onde é possível visualizar os documentos analíticos criados pelo utilizador. Destacado a funcionalidade de agrupar os documentos analíticos por tipo. O relatório "Voos_Atrasos_2015" onde é possível partilhar (*{U.C.3.1.3} Share Reports*), usar (*{U.C.3.1.2} Use Reports*), apagar (*{U.C.3.1.4} Delete Reports*) e alterar (*{U.C.3.1.5} Update Reports*). Estas funcionalidades estão disponíveis em todos os documentos analíticos criados e ao qual se tenha acesso.

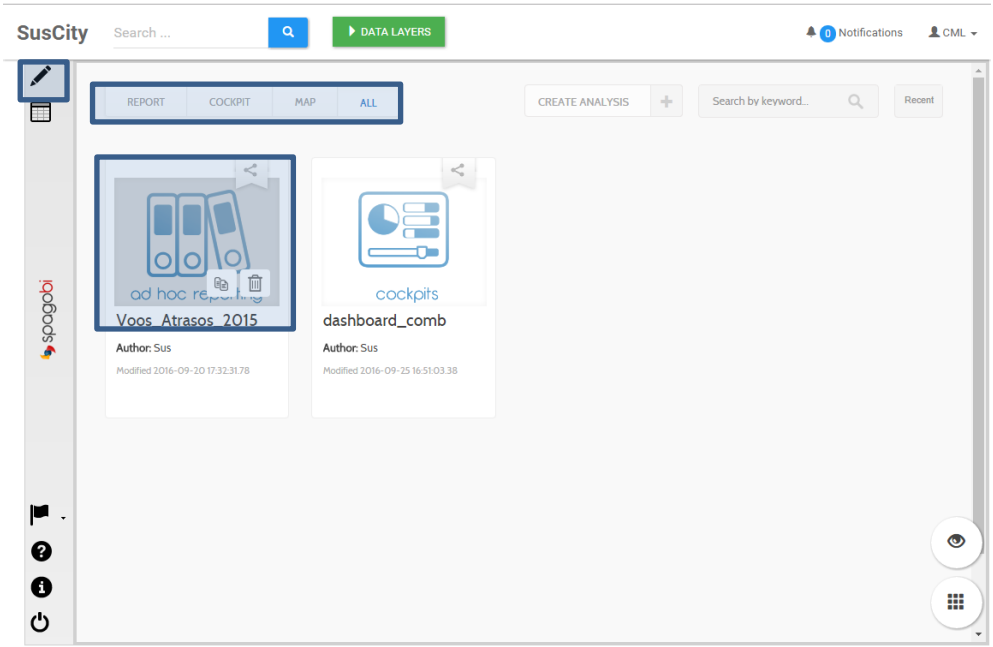


Figura 48 - Funcionalidades de *My Analysis.*

Na Figura 49, estão destacadas as funcionalidades de criar os diversos tipos de documentos analíticos. Existe a funcionalidade de *ad hoc reporting* (*{U.C.3.1.1} Create Report)*, *location int.* (*{U.C.3.3.1} Create Map*) e *cockpit* (*{U.C.3.2.1} Create Dashboard).*

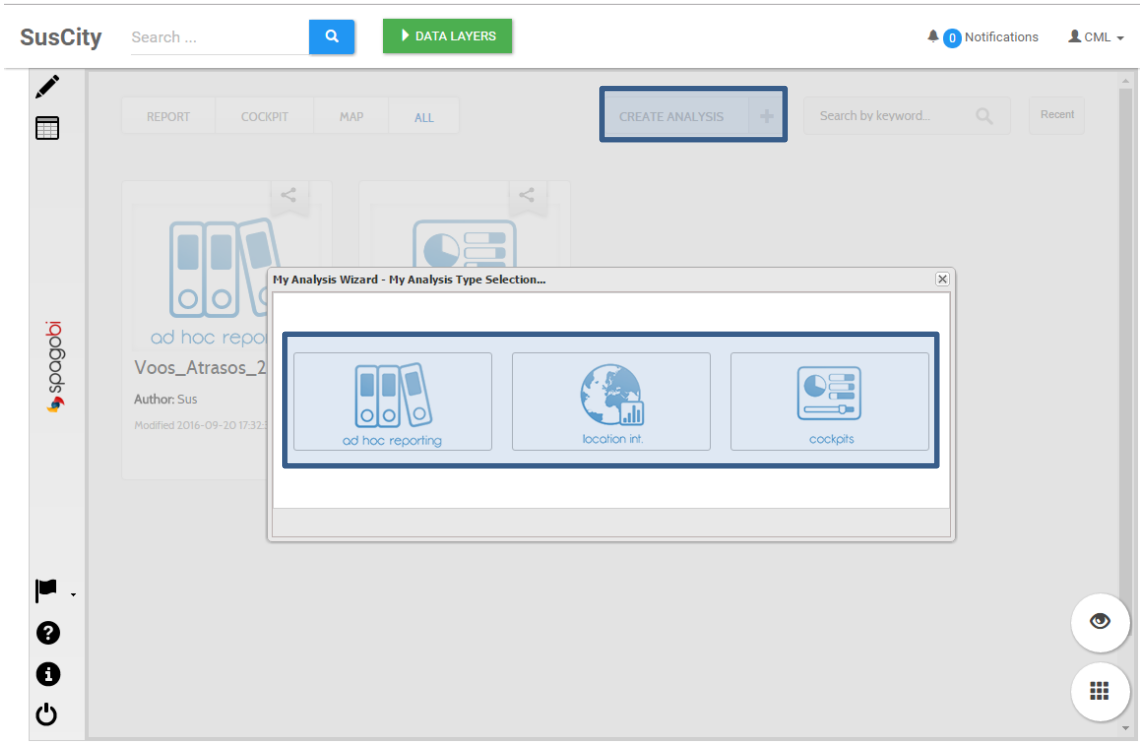


Figura 49 - Funcionalidade de *Create Analysis.*

Depois de apresentadas as funcionalidades disponíveis na plataforma, de seguida são demonstrados dois exemplos da utilização do serviço.

4.3.1. Utilização do serviço de *Upload* e análise de ficheiros

Como descrito na arquitetura o cidadão poderá fazer *upload* de um qualquer ficheiro *xls* ou *csv* que pretenda analisar. Tecnologicamente, e por restrições temporais, não foi possível o desenvolvimento da funcionalidade de *upload* de ficheiros diretamente para o *HDFS*, no entanto para demonstração foi utilizado o serviço base do *SpagoBI Server*, que guarda estes ficheiros numa localização interna da plataforma.

Para demonstração desta funcionalidade foi utilizado um ficheiro *xls* que contém dados de consumos de combustível automóvel nos últimos 5 anos. Este ficheiro contém dados distribuídos por ano, mês e valor do consumo em euros. Como visível na Figura 50, temos disponível a funcionalidade de *upload* de ficheiros.

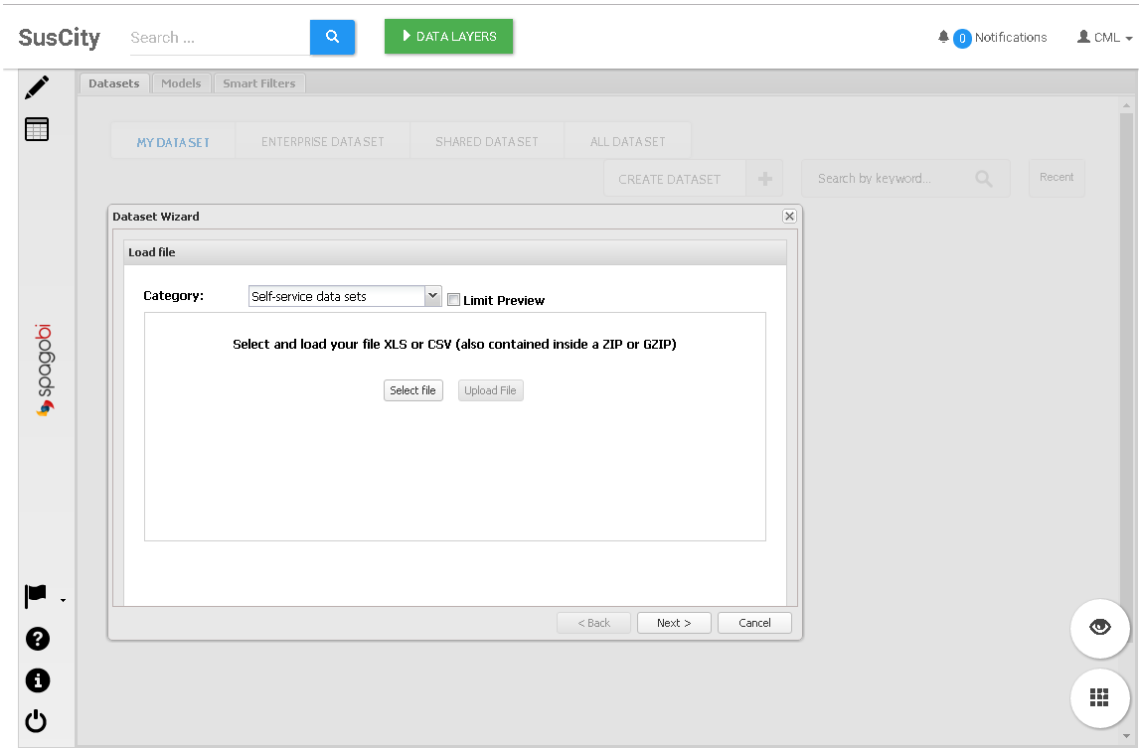


Figura 50 - Exemplo de *upload* de ficheiro.

Como descrito na subsecção 3.3.2, e depois de escolher os campos que serão considerados como atributos e os que serão considerados como métricas o *dataset* fica disponível para análise, como visível na Figura 47. No exemplo seguinte foi usado a funcionalidade de criar relatórios, construindo um gráfico que pode ser analisado por ano e mês. A título de exemplo, foi analisado com auxílio do gráfico criado, visível na Figura 51, o mês 8, referente ao mês de agosto, onde podemos verificar que o valor gasto em euros tem oscilado entre os 150€ no ano de 2013 e um pouco abaixo dos 125€ nos anos de 2014 e 2015.

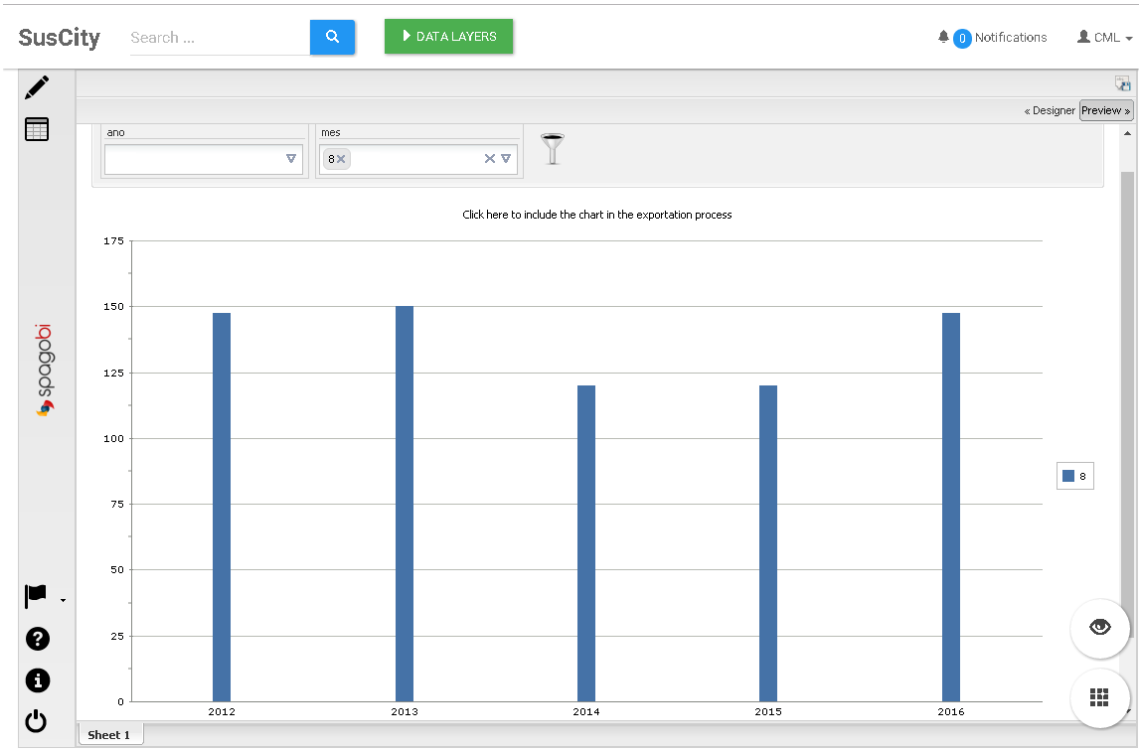


Figura 51 - Análise ao consumo combustível no mês de agosto.

Com a funcionalidade de *dashboard* podemos visualizar os dados em diversos tipos de gráficos e de tabelas, em diversas perspetivas. Como exemplo foi usado o *dataset* dos consumos de combustível e foi construído um *dashboard* visível na Figura 52 com diversos tipos de gráficos.

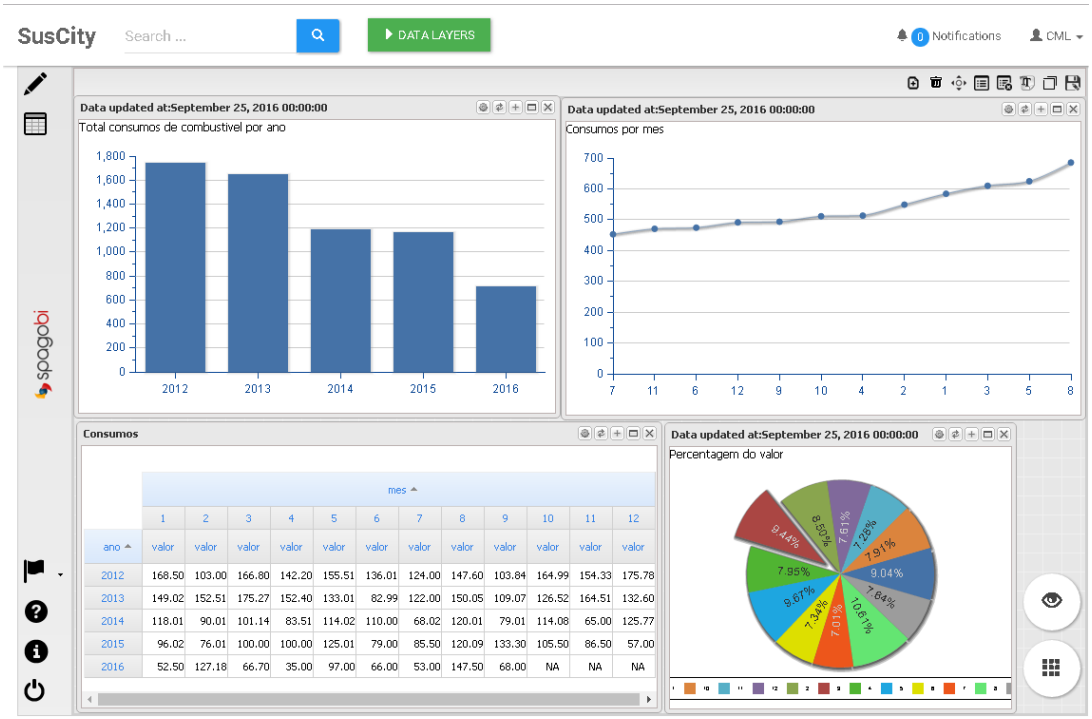


Figura 52 - *Dashboard* consumos de combustível.

4.3.2. Utilização da funcionalidade *ad hoc queries* em *BDW*

Como descrito na secção 4.3 a tabela Voos, ficou disponível na plataforma *Cloudera*. Sobre esta tabela, resultante da materialização da abordagem anteriormente descrita, foi construído o acesso aos dados da *smart city* no *SpagoBI Meta*. Este acesso foi exportado para a plataforma *SpagoBI Server* ficando disponível para ser usado nas análises. Como visível na Figura 53 o acesso a dados da *smart city* pode ser acedido e explorado na plataforma "SusCity".

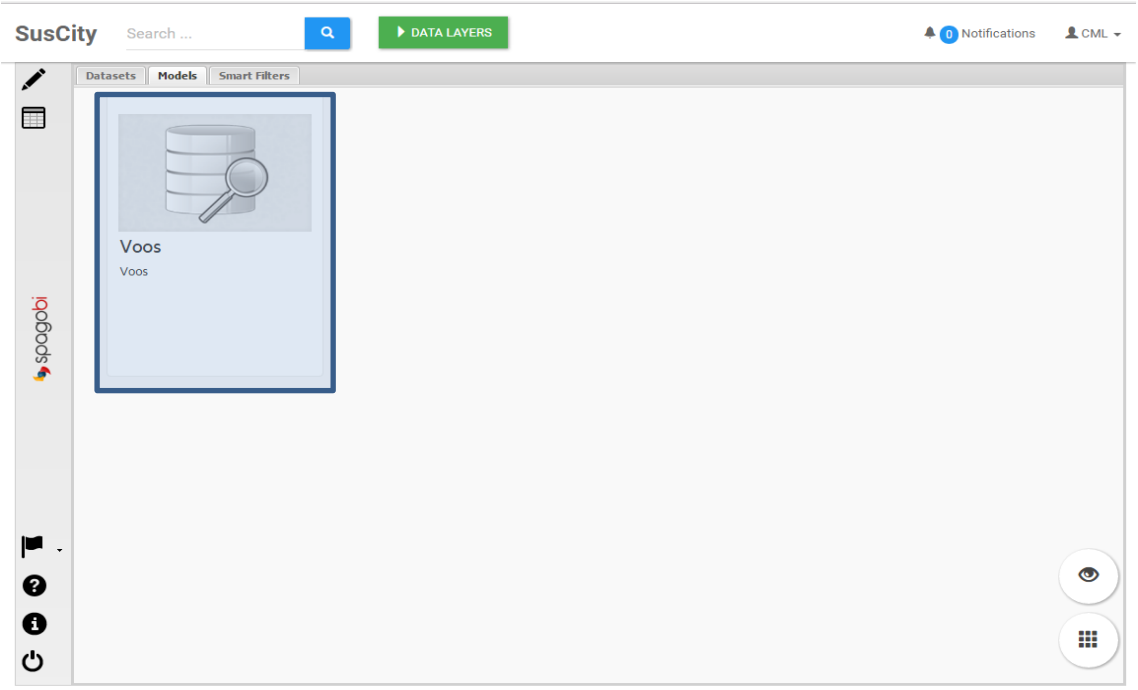


Figura 53 - Exemplo de acesso a dados da *smart city* disponibilizado.

Como descrito anteriormente na definição da arquitetura, este acesso aos dados da *smart city* disponibilizado ao cidadão, permite que este possa fazer o arrasto dos campos de uma forma *ad hoc* e responder a diversas questões. A título de exemplo, e supondo que o cidadão pretende saber a quantidade de voos que foram realizados por ano nos Estados Unidos e observar esse resultado num gráfico.

Na Figura 54 podemos observar os campos que foram arrastados e que definem a resposta à questão anterior.

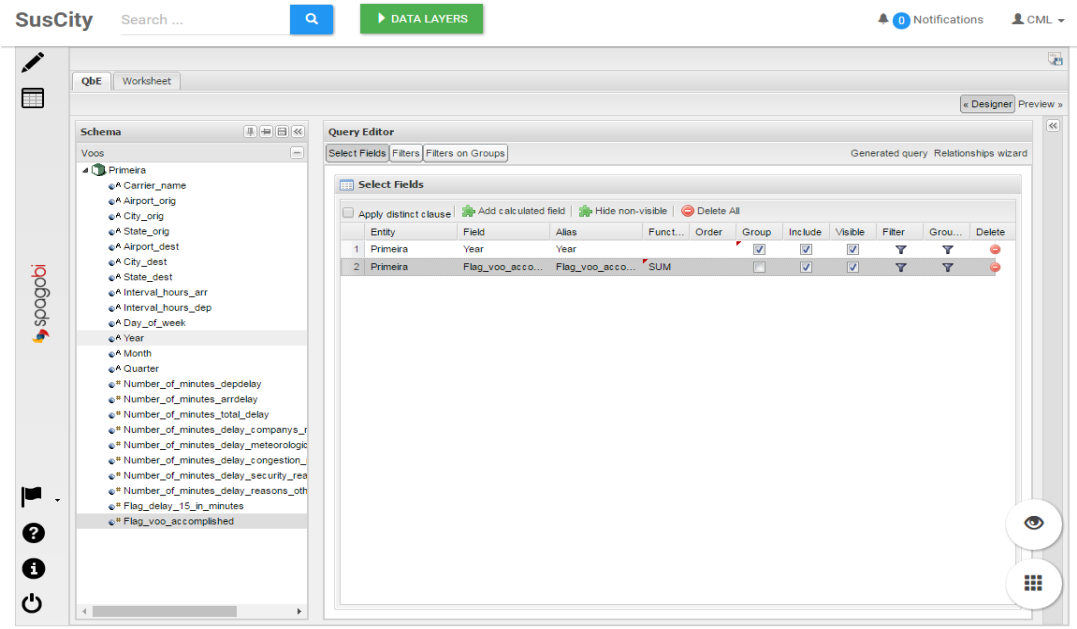


Figura 54 - Exemplo *ad hoc queries* do número de voos por ano.

Na Figura 55, encontramos o gráfico que representa a resposta a questão anterior, onde podemos ver o número de voos feitos, e a sua evolução ao longo do tempo.

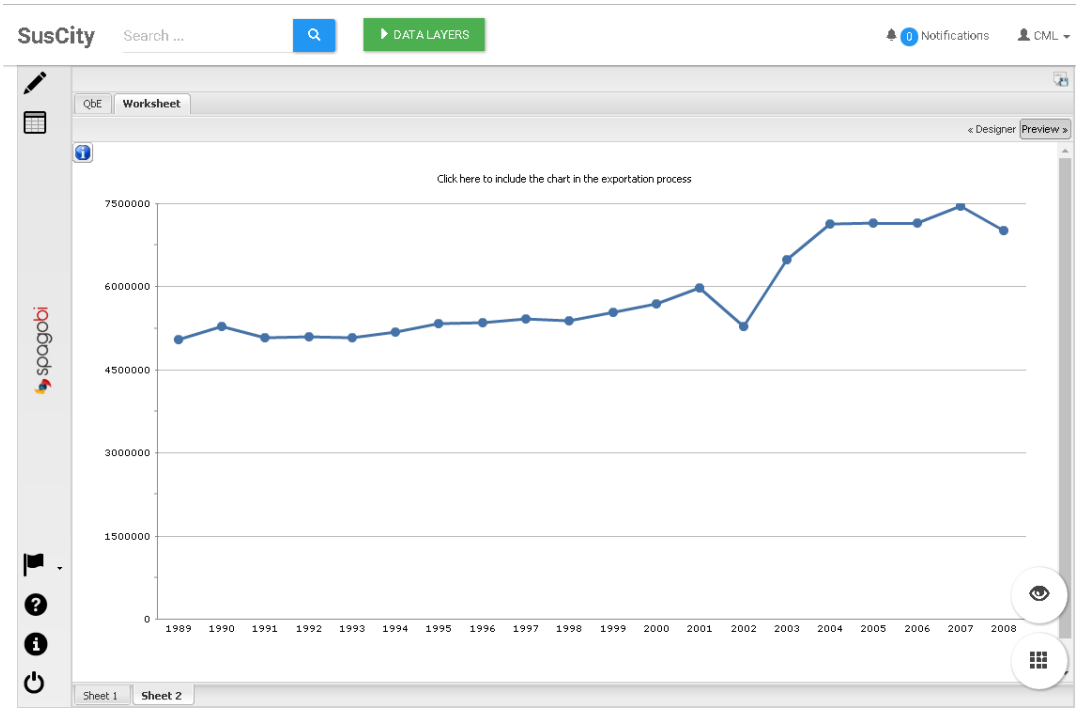
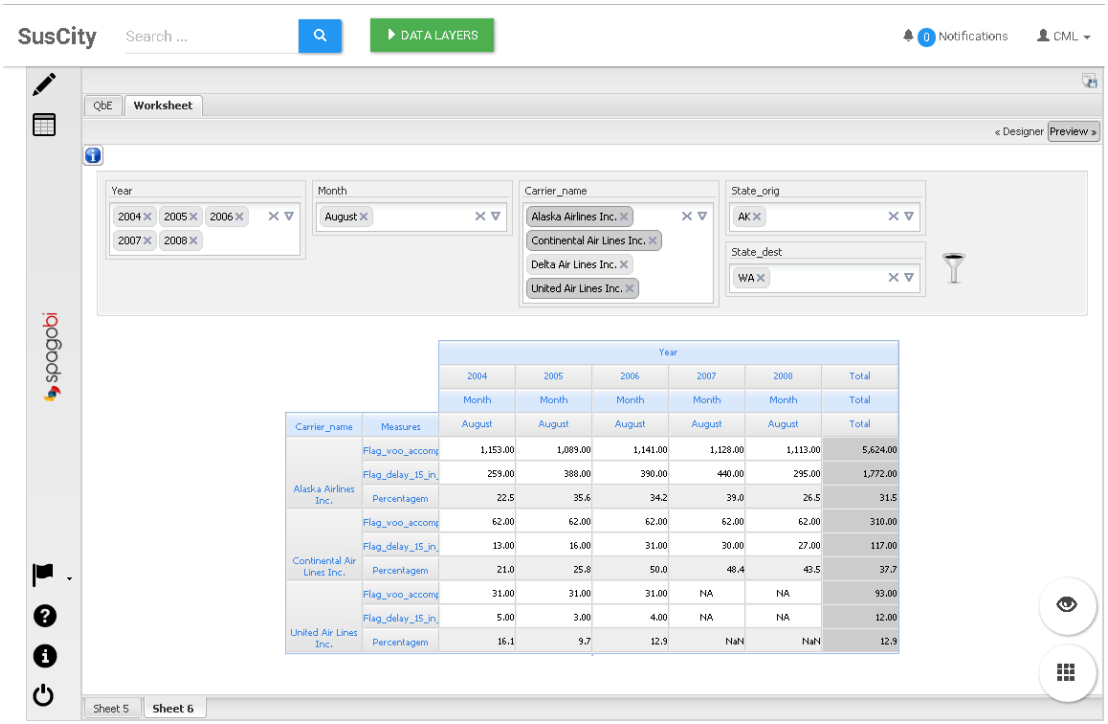


Figura 55 - Gráfico do número de voos por ano.

Supondo que o cidadão pretende voar no mês de agosto, e pretende saber quais as companhias que fazem voos do estado de origem *Alaska* e o de destino *Washington*, e destas

qual a que tem atrasos inferiores a 15 minutos em relação a totalidade de voos realizados nos últimos 5 anos.

Na Figura 56, podemos verificar que apenas três companhias fazem esta rota e que uma delas já não faz voos desde 2006. Foi ainda calculado um campo adicional, percentagem, que ajuda a perceber qual a companhia que ao longo dos anos tem a percentagem mais baixa de atrasos em relação à totalidade dos voos. Verifica-se que a companhia que tem melhores resultados é a *United Air Lines* mas que já não faz voos para estes destinos. Podemos concluir que a que melhor responde a questão colocada é a a *Alaska Air Lines*.



Carrier_name	Measures	2004 Month August	2005 Month August	2006 Month August	2007 Month August	2008 Month August	Total Month August
Alaska Airlines Inc.	Flag_voo_accomp	1,153.00	1,089.00	1,141.00	1,128.00	1,113.00	5,624.00
	Flag_delay_15_in_	259.00	388.00	390.00	440.00	295.00	1,772.00
	Percentagem	22.5	35.6	34.2	39.0	26.5	31.5
Continental Air Lines Inc.	Flag_voo_accomp	62.00	62.00	62.00	62.00	62.00	310.00
	Flag_delay_15_in_	13.00	16.00	31.00	30.00	27.00	117.00
	Percentagem	21.0	25.8	50.0	48.4	43.5	37.7
United Air Lines Inc.	Flag_voo_accomp	31.00	31.00	31.00	NA	NA	93.00
	Flag_delay_15_in_	5.00	3.00	4.00	NA	NA	12.00
	Percentagem	16.1	9.7	12.9	NaN	NaN	12.9

Figura 56 – Análise de voos *Alaska-Washinton* nos últimos 5 anos.

Nestes dois exemplos, e com os recursos computacionais já descritos, os tempos de *MapReduce* variam entre os treze e os quinze minutos.

4.4. Avaliação da arquitetura

Esta proposta de arquitetura foi desenvolvida tendo por base os conceitos *DAaaS*, providenciando uma infraestrutura completamente baseada na *cloud* e oferecida no paradigma *as-a-Service.* Conceptualmente os componentes propostos foram pensados de uma forma abrangente com o objetivo de colmatar as necessidades analíticas de uma *Smart City*, providenciando o detalhe necessário para a sua correta interpretação.

Tecnologicamente a arquitetura definida neste trabalho foi testada através da validação e implementação dos componentes destacados na Figura 44, em contexto de *Big Data*. Não

obstante da arquitetura ser validada em contexto real, esta está definida para que possa ser explorada e implementada seguindo as linhas gerais apresentadas.

Depois de realizados os dois exemplos anteriormente descritos com sucesso, verifica-se que a plataforma *SpagoBI* se enquadra na resolução do problema proposto apesar das limitações encontradas ao nível tecnológico. Estas limitações foram encontradas ao longo da experimentação e validação da arquitetura e serão descritas em pormenor nas conclusões desta dissertação. De destacar a funcionalidade *ad hoc queries* que se revelou inviável quando determinada query conduz a agregações de elevado número (mais de 500000 linhas), e ainda os tempos de resposta elevados do *MapReduce* para a disponibilização dos resultados das *queries,* sendo este o principal problema encontrado na experimentação realizada.

Da validação feita ao serviço de *upload* podemos constatar que este, sendo oferecido no paradigma *as-a-Service,* permite a qualquer utilizador fazer o *upload* de ficheiros e fazer as suas respetivas análises e ainda contribuir para o aumento do volume dos dados disponíveis. Nativamente o *SpagoBI*, guarda estes ficheiros numa diretoria da sua instalação pelo que se considera necessário o desenvolvimento de uma funcionalidade que permita o *upload* e leitura de ficheiros diretamente no *HDFS*.

Quanto à experimentação realizada em contexto *Big Data*, esta revelou que para além da sua utilidade, quando agregado ao componente proposto de *Inquiry,* dá a possibilidade a qualquer utilizador para fazer as suas análises de uma forma livre e retirar valor dos dados disponíveis. O problema encontrado foi a desempenho. O uso do *MapReduce* leva a que os tempos de resposta sejam elevados. Como descrito na subsecção 4.1.2, aconselha-se o desenvolvimento de suporte, na plataforma *SpagoBI Server*, para os *drivers* que tenham por base os motores mais ágeis, como o *Impala* e/ou *Spark* e/ou *Storm*, por forma a agilizar o processamento e por consequência a visualização de grandes volumes de dados. Visto que o *SpagoBI* tem suporte *Hive*, a utilização do motor *Tez* poderá ser uma solução a ponderar visto que este promete melhor desempenho que o *MapReduce.*

Podemos então concluir em relação à arquitetura apresentada, que a sua validação na plataforma *Cloudera*, *SpagoBI* e a sua integração no portal "SusCity", esta demonstra ser abrangente ao nível analítico funcionando bem, desde que os dados sejam disponibilizados de forma rápida, e que apesar do *SpagoBI* ter capacidade de processamento analítico, idealmente terá de se utilizar motores mais ágeis.

5.CONCLUSÃO

Esta dissertação enquadra o tema *Analytics-as-a-Service* no contexto de plataformas de *Big Data* para *Smart Cities*, em duas perspetivas, a conceptual e a tecnológica. Define uma proposta de arquitetura *DAaaS* que supra as necessidades analíticas de uma *Smart City* através de uma proposta de detalhe analítico à arquitetura BASIS. Para finalizar foram tecidas conclusões, apontando os pontos desenvolvidos, as principais limitações encontradas e, ainda, propostas de trabalho futuro.

5.1. Trabalho realizado

Esta dissertação começa por enquadrar conceptualmente o tema e define as principais motivações que levaram à sua realização. É exposta a abordagem metodológica utilizada, assim como a finalidade e principais objetivos a atingir.

Seguidamente introduz-se os conceitos necessários ao entendimento do *Big Data*, sendo ainda apresentadas as arquiteturas usadas pela IBM e Oracle nesta temática. No que diz respeito ao *Big Data Analytics*, são apresentados os conceitos gerais sobre o termo *analytics* e a sua aplicabilidade. Analisando diversos exemplos verificou-se tanto a utilidade, como a mais-valia retirada da implementação de contextos analíticos recorrendo a tecnologias *Big Data*.

Posteriormente é apresentado o *DAaaS (Data Analytics-as-a-Service)* como conceito necessário ao desenvolvimento de plataformas que sejam capazes de disponibilizar um componente analítico como um serviço. São ainda retratados os componentes necessários à construção de uma plataforma deste tipo, bem como apresentada uma visão de arquitetura neste paradigma. Ainda neste contexto são referidos os conceitos do *Data Analytics* para *Smart Cities* e são descritos alguns exemplos de contextos analíticos já desenvolvidos em algumas cidades, onde diversos autores demonstram a necessidade de aprofundar, melhorar e definir a componente analítica.

No que diz respeito ao enquadramento tecnológico, é apresentada a arquitetura BASIS como sendo o ponto de partida, onde irá assentar a arquitetura que esta dissertação irá propor como solução de base analítica para uma plataforma deste tipo. Para a concretização da arquitetura proposta foram comparadas as plataformas *Pentaho*, *SpagoBI*, *BIRT* e *Jaspersoft*, tendo-se concluído que a plataforma *SpagoBI* é a que melhor se enquadra ao trabalho a realizar. Após a validação da plataforma *SpagoBI* através de pequenos exemplos, esta demonstrou

capacidade para suprir parte das necessidades encontradas no ambiente analítico. Das conclusões retiradas através da validação à plataforma *SpagoBI*, ficou claro que nem todas as ferramentas necessárias ao desenvolvimento e materialização desta proposta estão disponíveis no paradigma *as-a-Service.*

Pela necessidade de uma arquitetura que, do ponto de vista conceptual e tecnológico, retratasse as temáticas necessárias ao contexto *Analytics-as-a-Service*, foi proposto o detalhe analítico para a arquitetura BASIS, bem como a descrição de todos os componentes propostos. Foram ainda formalmente descritas as principais funcionalidades do *DAaaS* através da utilização de UML, nomeadamente casos de uso e suas descrições. Como prova de conceito desta arquitetura foi, ainda que parcialmente, testada com sucesso a integração da plataforma *SpagoBI* com a plataforma "SusCity" por forma a disponibilizar serviços analíticos aos cidadãos. Da experimentação realizada identificaram-se as principais limitações da plataforma *SpagoBI*, bem como as da utilização do *MapReduce*.

5.2. Limitações

A utilização de acessos a dados da *Smart City* em *OLAP* e a utilização da funcionalidade *ad hoc queries* onde constam milhões de registos demonstrou-se inviável em determinadas circunstâncias, na medida em que o tempo de processamento com o *MapReduce,* que determinada *query* pode levar, por vezes tornar-se excessivo. Visto que a plataforma *Cloudera* usa os motores *MapReduce* e *Impala,* e o *SpagoBI* não suporta o *Impala*, foram notórios os problemas de desempenho. Foi constatado que determinadas *queries* consomem muito tempo até ser disponibilizado o seu resultado, podendo originar o evento de *timeout* no *SpagoBI*. A configuração deste evento está em trinta minutos, tempo que se considera excessivo, por exemplo, quando estamos à espera de resultados para construir um gráfico. Existe, ainda, a impossibilidade de exibição de resultados pela plataforma *SpagoBI,* quando determinada *query* conduz a agregações de elevado número (mais de 500000 linhas). Esta limitação pode ser ultrapassada através da parametrização dos ficheiros de configuração. Nativamente o *SpagoBI Server* é executado em memória *RAM*, quanto maior for o volume de linhas de um determinado contexto de análise, maior será a alocação de memória *RAM*. Os testes executados neste contexto, e com as limitações de *hardware* existentes, levaram a concluir que a partir dos 6GB de *RAM* o motor demonstra instabilidade deixando mesmo de responder. Também se observou que quando os resultados obtidos tanto em cubos *OLAP* como na funcionalidade *ad hoc queries* são inferiores a 100000 registos o *SpagoBI* consegue responder em alguns segundos.

Uma outa limitação encontrada foi o *hardware*, pois o trabalho foi realizado em máquinas virtuais, com as caraterísticas já referidas anteriormente, o que tornou o processo mais moroso.

Por último constatou-se que a comunidade *SpagoBI* é pouco ativa na resolução de problemas, sendo que muitas questões que surgiram ao longo desta dissertação já se encontravam colocadas, algumas delas com anos, e sem qualquer tipo de resposta por parte da organização responsável do *software*.

5.3. Trabalho futuro

Após a conclusão desta dissertação existem alguns pontos da arquitetura que necessitam de ser validados e aprofundados para uma implementação bem-sucedida em contexto real.

Identifica-se a necessita de investigar qual, ou quais os motores de processamento mais indicados, servindo especificamente os componentes analíticos propostos. Aconselha-se a validação da plataforma *SpagoBI* com o motor *Tez* visto que podem ser usados os mesmos *drivers* do *Hive* ao qual o *SpagoBI* já tem suporte. Motores como o *Storm, Spark* e *Impala* são destacados na arquitetura, no entanto o *SpagoBI* não os suporta, pelo que a sua validação terá primeiramente de ser preparada através da sua implementação no *SpagoBI*.

No que diz respeito às ferramentas que necessitam do seu desenvolvimento no paradigma *as-a-Service* identifica-se o *Spago Meta* para a criação de acessos a dados da *Smart City* de forma *ad hoc,* e o *Spago Studio* para a definição de *KPIs* e relatórios avançados.

É necessário desenvolver a capacidade no *SpagoBI* de ler e escrever os ficheiros que o cidadão faz *upload,* diretamente para o *HDFS*.

Por último, e como já identificado nesta dissertação, se o ambiente de aplicação assim o exigir, será necessário migrar a base de dados operacional do *SpagoBI* que se encontra em *HyperSQL* para o paradigma *Big Data.*

REFERÊNCIAS BIBLIOGRÁFICAS

Arnold, K., & Campbell, J. (2011). Course Signals: A Student Success System/Stoplights for Student Success Retrieved from http://www.educause.edu/annual-conference/2011/course-signals-student-success-systemstoplights-student-success

Atos. (2013). Data Analytics as a Service: unleashing the power of Cloud and Big Data. Retrieved from http://docplayer.net/307657-Paper-white-ascent-data-analytics-as-a-service-unleashing-the-power-of-cloud-and-big-data-thought-leadership-from-atos.html

Barga, R. S., Ekanayake, J., & Lu, W. (2012). *Project daytona: Data analytics as a cloud service.* Paper presented at the Data Engineering (ICDE), 2012 IEEE 28th International Conference on.

Bernabei, A. (2014, 28-10-2014). SpagoBI Server Architecture. Retrieved from http://wiki.spagobi.org/xwiki/bin/view/spagobi_server/analytical_angines

Bin, C., Longo, S., Cirillo, F., Bauer, M., & Kovacs, E. (2015, June 27 2015-July 2 2015). *Building a Big Data Platform for Smart Cities: Experience and Lessons from Santander.* Paper presented at the Big Data (BigData Congress), 2015 IEEE International Congress on.

Boulos, M. N. K. (2015). Social, innovative and smart cities are happy and resilient. In A. D. Tsouros (Ed.), *insights from the WHO EURO 2014 International Healthy Cities Conference*: *International Journal of Health Geographics* 2015International Journal of Health Geographics 2015.

Bronstein, Z. (2009). Industry and the smart city. *Dissent, 56*(3), 27-34.

Costa, C. (2015). BASIS: Uma Arquitetura de Big Data para

Smart Cities. In (pp. 116). Dissertação de Mestrado, Universidade do Minho, 2015.

Demchenko, Y., Grosso, P., Laat, C. d., & Membrey, P. (2013, 20-24 May 2013). *Addressing big data issues in Scientific Data Infrastructure.* Paper presented at the 2013 International Conference on Collaboration Technologies and Systems (CTS).

Dodson, V. (2013). Visualizing Big Data with Hadoop and BIRT. Retrieved from http://developer.actuate.com/community/forum/index.php/blog/10/entry-479-visualizing-big-data-with-hadoop-and-birt/

Eclipse. (2016). BIRT Eclipse Demo. Retrieved from http://demo.actuate.com/demos/EclipseDemo/EclipseDemo.html

Gibson, D. V., Kozmetsky, G., & Smilor, R. W. (1992). *The technopolis phenomenon: Smart cities, fast systems, global networks*: Rowman & Littlefield.

Giffinger, R., & Gudrun, H. (2010). Smart cities ranking: an effective instrument for the positioning of the cities? *ACE: Architecture, City and Environment, 4*(12), 7-26.

Groenfeldt, T. (2012). Using Big Data to Help A Hospital Meet The Financial Future. Retrieved from http://www.forbes.com/sites/tomgroenfeldt/2012/04/20/aurora-health-uses-big-data-to-reduce-risk-in-outcomes-based-pay/

Hohpe, G., & Woolf, B. (2004). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*: Addison-Wesley.

Huang, J., Zhang, R., Buyya, R., Chen, J., & Wu, Y. (2016). Heads-Join: Efficient Earth Mover's Distance Similarity Joins on Hadoop. *IEEE Transactions on Parallel and Distributed Systems, 27*(6), 1660-1673. doi:10.1109/TPDS.2015.2462354

Hugh, J. W. (2013). All About Analytics. *International Journal of Business Intelligence Research (IJBIR), 4*(1), 13-28. doi:10.4018/jbir.2013010102

IBM. (2012). Managing big data for smart grids and smart meterss. Retrieved from http://www-935.ibm.com/services/multimedia/Managing_big_data_for_smart_grids_and_smart_meters.pdf

Ishwarappa, & Anuradha, J. (2015). A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology. *International Conference on Computer, Communication and Convergence (Iccc 2015), 48*, 319-324. doi:10.1016/j.procs.2015.04.188

Jara, A. J., Genoud, D., & Bocchi, Y. (2014). Big data for smart cities with KNIME a real experience in the SmartSantander testbed. *Software: Practice and Experience*.

Jaradat, M., Jarrah, M., Bousselham, A., Jararweh, Y., & Al-Ayyoub, M. (2015). *The internet of energy: Smart sensor networks and big data management for smart grid.* Paper presented at the Procedia Computer Science.

Jaspersoft. (2015a). Community Wiki. Retrieved from http://community.jaspersoft.com/wiki/community-wiki

Jaspersoft. (2015b). Native reporting for Hadoop Hive, Impala and HBase with Jaspersoft Studio, iReport and JasperReports Server. Retrieved from http://community.jaspersoft.com/project/hadoop-connectors

Khan, Anjum, A., & Kiani, S. L. (2013). *Cloud based Big Data Analytics for Smart Future Cities*. Paper presented at the IEEE/ACM 6th International Conference on Utility and Cloud Computing.

Khan, Anjum, A., Soomro, K., & Tahir, M. A. (2015). Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing 2015Journal of Cloud Computing 2015*. doi:10.1186/s13677-015-0026-8

Khan, Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data Understanding Big Data to extract Value. *2014 Zone 1 Conference of the American Society for Engineering Education (Asee Zone 1)*, 5.

Klein, C. (2015). SpagoBI suite helps Sepaco Hospital in Sao Paulo, Brazil, to save lives. Retrieved from http://www.spagobi.org/2015/09/spagobi-suite-helps-sepaco-hospital-in-sao-paulo-brazil-to-save-lives/

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Retrieved from http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

Li, D., Shan, J., Shao, Z., Zhou, X., & Yao, Y. (2013). Geomatics for smart cities - concept, key techniques, and applications. *Geo-Spatial Information Science, 16*(1), 13-24. doi:10.1080/10095020.2013.772803

Martinho, B. (2016). Data Warehousing em contexto Big Data: Dos conceitos à implementação. In. Dissertação de Mestrado, Universidade do Minho, 2016.

Minneman, W. (1996). How to outsource a complex business process. *Hunter Group*.

Mitton, N., Papavassiliou, S., Puliafito, A., & Trivedi, K. S. (2012). Combining Cloud and sensors in a smart city environment. *EURASIP journal on Wireless Communications and Networking, 2012*(1), 1-10.

Nam, T., & Pardo, T. A. (2011). *Conceptualizing smart city with dimensions of technology, people, and institutions.* Paper presented at the Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times.

Ng, R., Arocena, P., Barbosa, D., & Carenini, G. (2013). *Perspectives on Business Intelligence*: Morgan & Claypool.

Noel, L., & Levitz, R. (2012). Retention Excellence Awards. Retrieved from https://www.ruffalonl.com/case-studies/student-retention-case-studies/retention-excellence-awards

Oracle. (2015a). An Enterprise Architect's Guide to Big Data. In *Reference Architecture Overview*: Oracle.

Oracle. (2015b). Oracle Health Sciences Network. Retrieved from http://www.oracle.com/us/products/applications/health-sciences/network/overview/index.html

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *J. Manage. Inf. Syst., 24*(3), 45-77. doi:10.2753/mis0742-1222240302

Pentaho. (2016). Documentation. Retrieved from https://help.pentaho.com/Documentation

Quintero, D., Cruz, L. C., Picone, R. M., Smolej, D., Casali, D. d. S., Tudor, G., & Wong, J. (2014). *IBM Platform Computing Solutions Reference Architectures and Best Practices* Redbooks.

Research, N. (2014). Analytics pays back $13.01 for every dollar spent. Retrieved from http://nucleusresearch.com/research/single/analytics-pays-back-13-01-for-every-dollar-spent/

RIBA-BTS. (2015). Bareau of Transportation Statistics. Retrieved from http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time

Rigsby, J. (2012). EMC Partners with Purdue University to Solve Big Data Problems Retrieved from http://siliconangle.com/blog/2012/07/09/emc-partners-with-purdue-university-to-solve-big-data-problems/

Rijmenam, M. v. (2014). *Think Bigger: Developing a Successful Big Data Strategy for Your Business*: Amacom.

Sanchez, L., Galache, J. A., Gutierrez, V., Hernandez, J. M., Bernat, J., Gluhak, A., & Garcia, T. (2011). *Smartsantander: The meeting point between future internet research and experimentation and the smart cities.* Paper presented at the Future Network & Mobile Summit (FutureNetw), 2011.

Santos, M. Y., & Costa, C. (2016). Data Warehousing in Big Data: From Multidimensional to Tabular Data Models. In. *Ninth International C* Conference on Computer Science & Software Engineering (accepted for publication)*. Porto, Portugal, 20-22 July 2016.

Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., & Oliveira, A. (2011). Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. In (pp. 431-446).

Schultz, B. (2012). Aurora & Others Collaborate on Information-Sharing Healthcare Network. Retrieved from http://www.allanalytics.com/author.asp?section_id=1411&doc_id=246903

Sessions, R. (2007). Comparison of the top four enterprise architecture methodologies. Retrieved from https://msdn.microsoft.com/en-us/library/bb466232.aspx

Shelton, T., Zook, M., & Wiig, A. (2015). The 'actually existing smart city'. *Cambridge Journal of Regions Economy and Society, 8*(1), 13-25. doi:10.1093/cjres/rsu026

SpagoBI. (2012). Business Intelligence With SpagoBI. In. Web: SpagoBI.

SpagoBI. (2014). Quick start. In *A project with SpagoBI 4* (V2 ed.): Engineering Group.

SpagoBI. (2016). FAQs. Retrieved from http://www.spagobi.org/homepage/faqs/

Suakanto, S., Supangkat, S. H., Suhardi, & Saragih, R. (2013, 13-14 June 2013). *Smart city dashboard for integrating various data of sensor networks.* Paper presented at the ICT for Smart Society (ICISS), 2013 International Conference on.

SusCity. (2016). Retrieved from http://groups.ist.utl.pt/suscity-project/inicio/

Tanko, M., & Burke, M. (2015). Why busways? Styles of planning and mode-choice decision-making in Brisbane's transport networks. *Australian Planner, 52*(3), 229-240. doi:10.1080/07293682.2015.1047873

Tarantola, A. (2013). How Prescriptive Analytics Could Harness Big Data to See the Future. Retrieved from http://gizmodo.com/how-prescriptive-analytics-could-harness-big-data-to-se-512396683

University, P. (2015). Course Signals. Retrieved from http://www.itap.purdue.edu/learning/tools/signals/

Watson, H. J. (2014). Tutorial: Big data analytics: Concepts, technologies, and applications. *Communications of the Association for Information Systems, 34*(1), 1247-1268.

Wenge, R., Zhang, X., Dave, C., Chao, L., & Hao, S. (2014). Smart city architecture: A technology guide for implementation and design challenges. *Communications, China, 11*(3), 56-69.

Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., & David, B. (2015). A literature survey on smart cities. *Science China Information Sciences, 58*(10), 1-18.

Young, E. (2015). *Becoming an Analytics-driven Organisation to Create Value*. Retrieved from http://www.ey.com/Publication/vwLUAssets/EY-becoming-an-analytics%E2%80%93driven-organisation-to-create-value/$FILE/EY-becoming-an-analytics%E2%80%93driven-organisation-to-create-value.pdf

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal, 26*(3), 276-292. doi:10.1147/sj.263.0276

Zhang, C., He, L., Mao, Y., & Xiao, B. (2015, 15-17 June 2015). *Knowledge discovery of network public opinion in the concept of smart city.* Paper presented at the Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on.

Zikopoulos, P., & Eaton, C. (2011). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*: McGraw-Hill Osborne Media.

ANEXOS

ANEXO A - Proposta de funcionalidades de *DAaaS*

Como apresentado na secção 4.2, este anexo detalha as principais funcionalidades propostas para o *DAaaS*.

O caso de uso seguinte, *{U.C.1} Manage Platform* contém três casos de uso folha como se pode verificar na Figura 57. A descrição deste caso de uso é apresentada abaixo na Tabela 13.

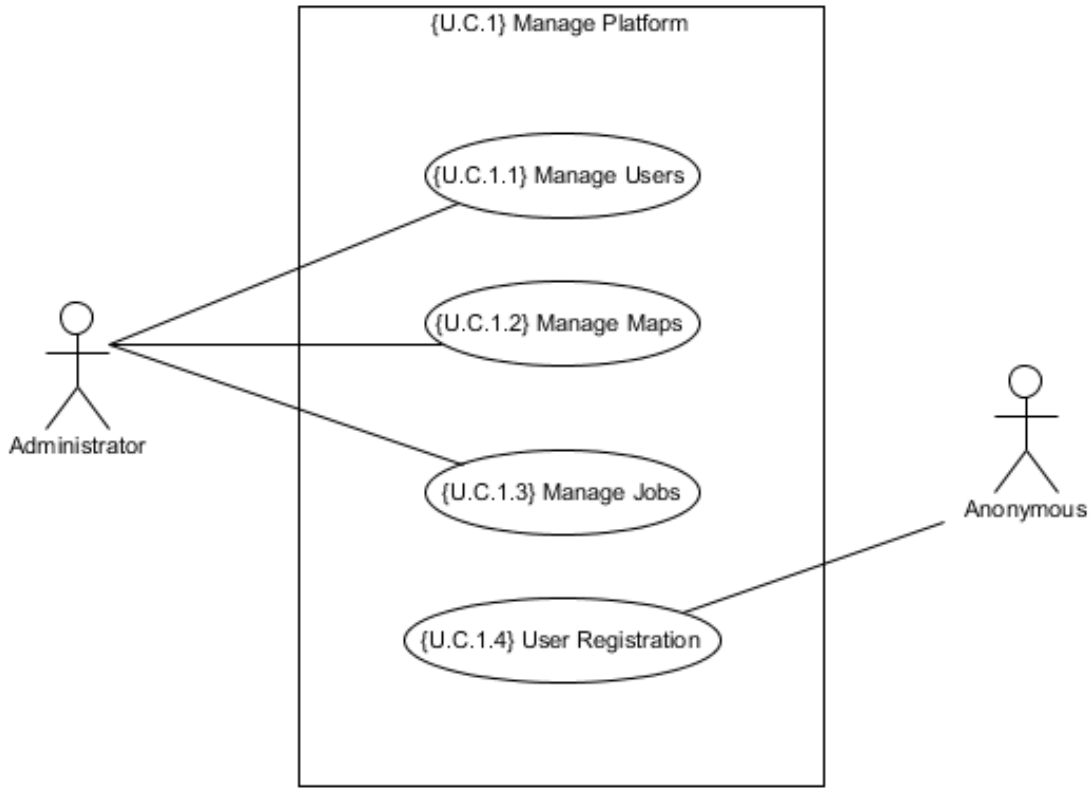


Figura 57 - *{U.C.1} Manage Platform.*

Tabela 13 - Especificação do caso de uso *{U.C.1} Manage Platform.*

{U.C.1} Manage Platform	
{U.C.1.1} Manage Users	Permite gerir os utilizadores da plataforma: • Funcionalidades de *CRUD (Create, read, update e delete)* de utilizadores; • *CRUD* de grupos de utilizadores; • Envio através de *e-logs* dos dados de acesso para novo utilizador (utilizador e password);

	• Alterar dados pessoais; • Definição de permissões de acesso aos diversos componentes analíticos; • Definição de restrições de uso, nomeadamente espaço em disco para *uploads* de ficheiros; • Número de trabalhos analíticos guardados por cada utilizador; • Listar utilizadores; • Pesquisar utilizadores.
{U.C.1.2} Manage Maps	Permite gerir os mapas: • Funcionalidades de *CRUD* em Mapas que servirão de base às análises, por exemplo, mapas de cidades; • Listar Mapas; • Pesquisar Mapas.
{U.C.1.3} Manage Jobs	Permite gerir os trabalhos agendados: • Funcionalidades de *CRUD* em *Jobs*; • Definição de tarefas automatizadas de gestão. Como por exemplo o aviso ao ator *Normal User* de datas limite para os seus ficheiros serem removidos ou integrados nas bases de dados da *Smart City*, um outro exemplo será o aviso ao ator *Normal User* que está preste a exceder a cota de espaço para os seus *uploads.*
{U.C.1.4} User Registration	Acesso a formulário de registo no sistema.

O caso de uso seguinte, *{U.C.2} Manage Smart City Data,* contém seis casos de uso folha como se pode verificar na Figura 58. A descrição deste caso de uso é apresentada abaixo na Tabela 14.

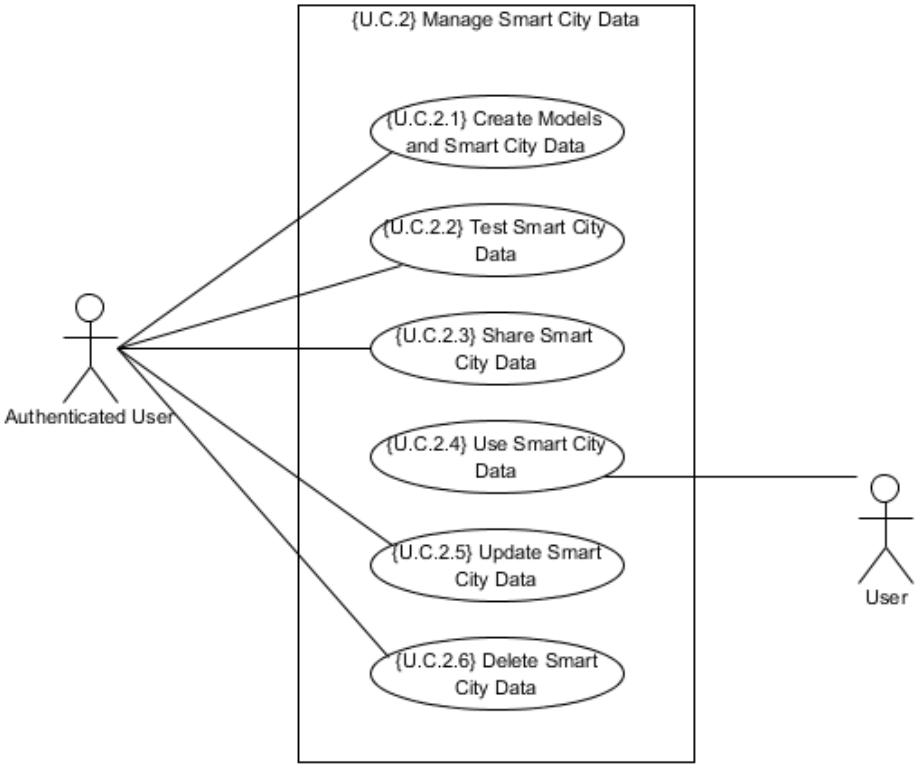


Figura 58 - *{U.C.2} Manage Smart City Data.*

Tabela 14 - Especificação do caso de uso *{U.C.2} Manage Smart City Data*.

{U.C.2} Manage Smart City Data	
{U.C.2.1} Create Models and Smart City Data	Permite criar modelos e acessos a dados da *Smart City:* • Permite criar acessos a dados da *Smart City*; • Permite definir cubos *OLAP*; • Permite definir campos considerados atributos e métricas.
{U.C.2.2} Test Smart City Data	Permite testar os acessos a dados da *Smart City*: • Testar em *ad hoc queries*; Testar usando funcionalidades descritas em {U.C.3}, {U.C.4} e {U.C.5}.
{U.C.2.3} Share Smart City Data	Permite partilhar os acessos a dados da *Smart City*: • Partilha com utilizadores específicos; • Partilha com grupos de utilizadores; • Partilha com todos os utilizadores; • Retirar a partilha nas funcionalidades anteriores.

Tabela 15 - Especificação do caso de uso *{U.C.2} Manage Smart City Data*.

{U.C.2} Manage Smart City Data	
{U.C.2.4} Use Smart City Data	Permite usar os acessos a dados da *Smart City*. • Usar a funcionalidade *ad hoc queries*; • Definir métricas com base em outras métricas utilizando operadores lógicos; • Agrupar por categorias. Após a seleção de dados da *Smart City*, torna-se possível usar as funcionalidades descritas em {U.C.3}, {U.C.4} e {U.C.5}.
{U.C.2.5} Update Smart City Data	Alterar acessos a dados da *Smart City.*
{U.C.2.6} Delete Smart City Data	Apagar acessos a dados da *Smart City.*

O caso de uso seguinte, *{U.C.3} Manage Analysis,* contém quatro casos de uso como se pode verificar na Figura 59. A descrição deste caso de uso é apresentada abaixo na Tabela 16.

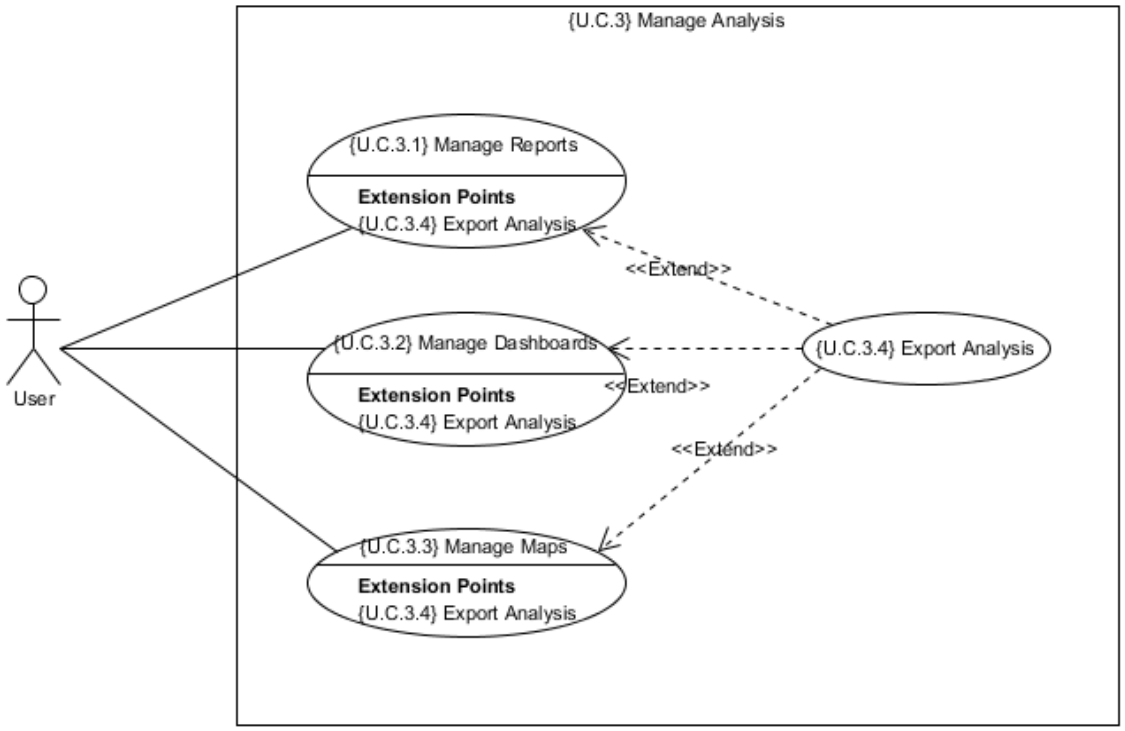


Figura 59 - *{U.C.3} Manage Analysis.*

Tabela 16 - Especificação do caso de uso *{U.C.3} Manage Analysis.*

{U.C.3} Manage Analysis.	
{U.C.3.1} Manage Reports	Permite gerir relatórios mediante permissões específicas.
{U.C.3.2} Manage Dashboards	Permite gerir *dashboards* mediante permissões específicas.
{U.C.3.3} Manage Maps	Permite gerir mapas mediante permissões específicas.
{U.C.3.4} Export Analysis	Permite exportar as análises em diversos formatos: • Exportar para *pdf*; • Exportar para *xls*; • Exportar para *xlsx*; • Exportar para *csv*; • Exportar para *jpeg*; • Imprimir.

O caso de uso seguinte, *{U.C.3.1} Manage Reports,* contém sete casos de uso folha como se pode verificar na Figura 60. A descrição deste caso de uso é apresentada abaixo na Tabela 17.

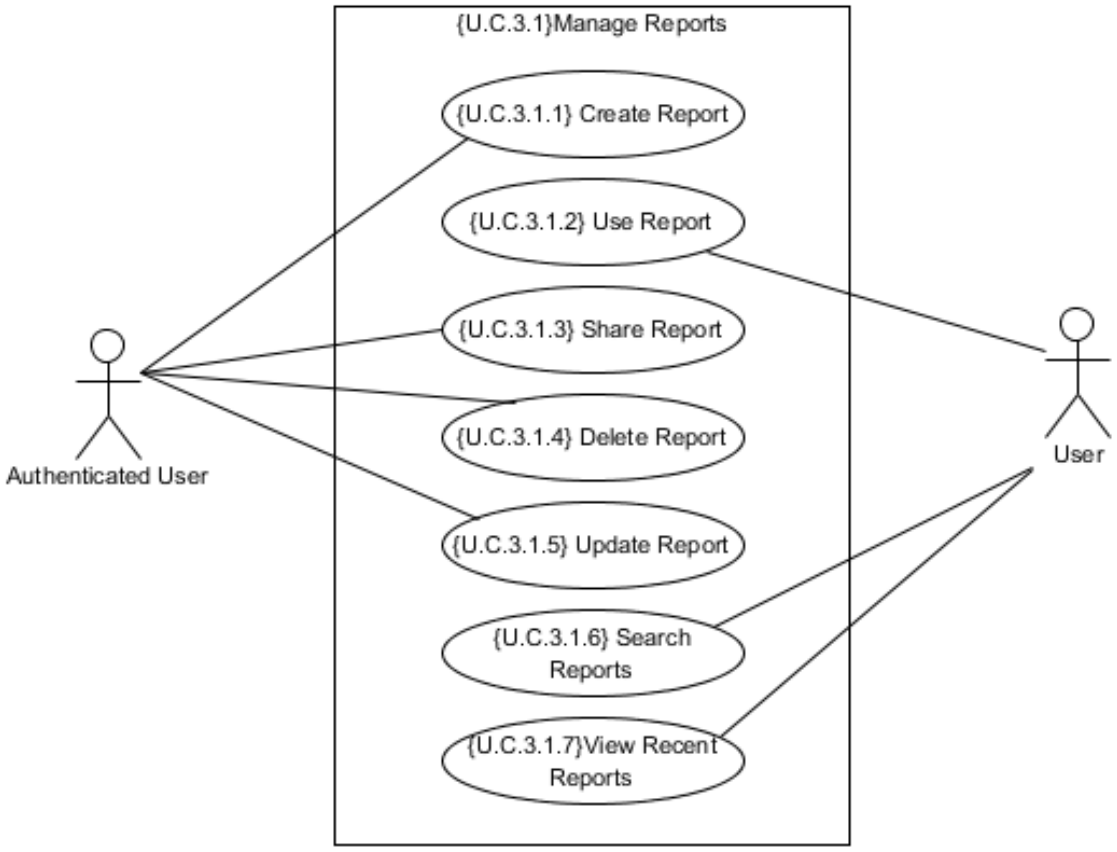


Figura 60 - *{U.C.3.1} Manage Reports.*

Tabela 17 - Especificação do caso de uso *{U.C.3.1} Manage Reports.*

{U.C.3.1} Manage Reports.	
{U.C.3.1.1} Create Report	Pré-condição: Foi previamente selecionado pelo menos um *dataset* ou dados da *Smart City*. Permite criar relatórios com base nos *datasets* disponíveis. Estes podem ser baseados em gráficos diversos e tabelas.
{U.C.3.1.2} Use Reports	Permite usar relatórios disponíveis e agrupar por categorias.
{U.C.3.1.3} Share Reports	Permite partilhar relatórios e remover a sua partilha. • Partilha de relatórios com grupos de utilizadores; • Partilha de relatórios com todos os utilizadores;
{U.C.3.1.4} Delete Reports	Permite remover relatórios.
{U.C.3.1.5} Update Reports	Permite atualizar relatórios.
{U.C.3.1.6} Search Reports	Permite procurar relatórios.
{U.C.3.1.7} Recent Reports	Permite visualizar relatórios recentemente usados.

O caso de uso seguinte, *{U.C.3.2} Manage Dashboards,* contém sete casos de uso folha como se pode verificar na Figura 61. A descrição deste caso de uso é apresentada abaixo na Tabela 18.

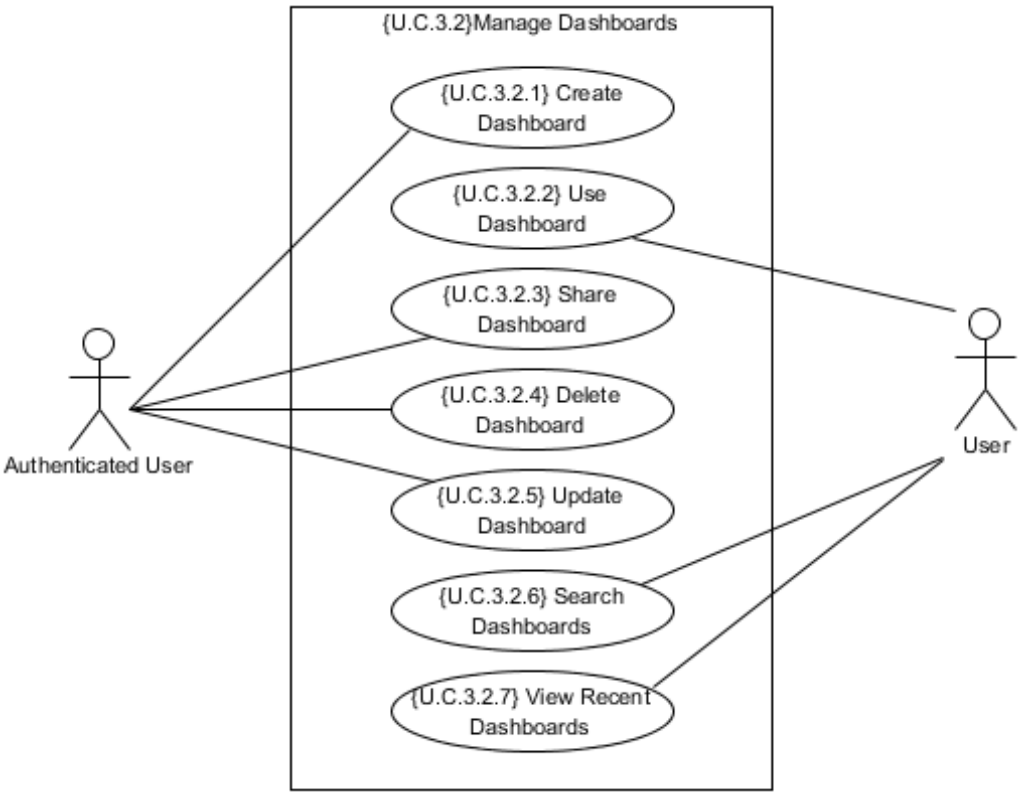


Figura 61 - *{U.C.3.2} Manage Dashboards.*

Tabela 18 - Especificação do caso de uso *{U.C.3.2} Manage Dashboards.*

{U.C.3.2} Manage Dashboards	
{U.C.3.2.1} Create Dashboard	Permite criar *dashboard* com base nos *datasets* e/ou dados da *Smart City* disponíveis. Estes podem ser baseados em gráficos diversos e tabelas.
{U.C.3.2.2} Use Dashboard	Permite usar os *dashboard* disponíveis e agrupar por categorias.
{U.C.3.2.3} Share Dashboard	Permite partilhar *dashboard* e remover a sua partilha. • Partilha com grupos de utilizadores; • Partilha com todos os utilizadores;
{U.C.3.2.4} Delete Dashboard	Permite remover *dashboard*.
{U.C.3.2.5} Update Dashboard	Permite atualizar *dashboard*.
{U.C.3.2.6} Search Dashboards	Permite procurar *dashboards*.
{U.C.3.2.7} Recent Dashboards	Permite visualizar dashboards recentemente usados.

O caso de uso seguinte, *{U.C.3.3} Manage Maps,* contém sete casos de uso folha como se pode verificar na Figura 62. A descrição deste caso de uso é apresentada abaixo na Tabela 19.

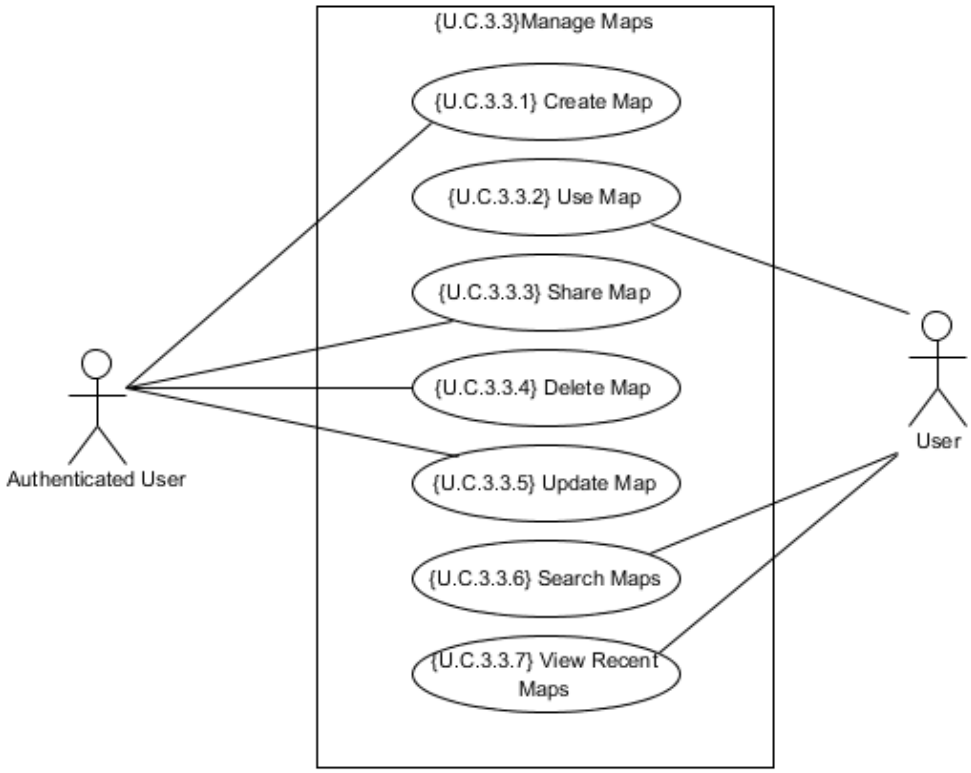


Figura 62 - *{U.C.3.3} Manage Maps.*

Tabela 19 - Especificação do caso de uso *{U.C.3.3} Manage Maps*.

{U.C.3.3} Manage Maps	
{U.C.3.3.1} Create Map	Pré-condição: Foi previamente selecionado pelo menos um *dataset* ou dados da *Smart City*. Permite criar análises inseridas em mapas.
{U.C.3.3.2} Use Map	Permite usar as análises disponíveis em mapas e agrupar por categorias.
{U.C.3.3.3} Share Map	Permite partilhar análises em mapas e remover a sua partilha. • Partilha com grupos de utilizadores; • Partilha com todos os utilizadores;
{U.C.3.3.4} Delete Map	Permite remover análises em mapas.
{U.C.3.3.5} Update Map	Permite atualizar análises em mapas.
{U.C.3.3.6} Search Maps	Permite procurar análises em mapas.
{U.C.3.3.7} Recent Maps	Permite visualizar relatórios recentemente usados.

O caso de uso seguinte, *{U.C.4} Manage Datasets* contém sete casos de uso folha como se pode verificar na Figura 63. A descrição deste caso de uso é apresentada abaixo na Tabela 20.

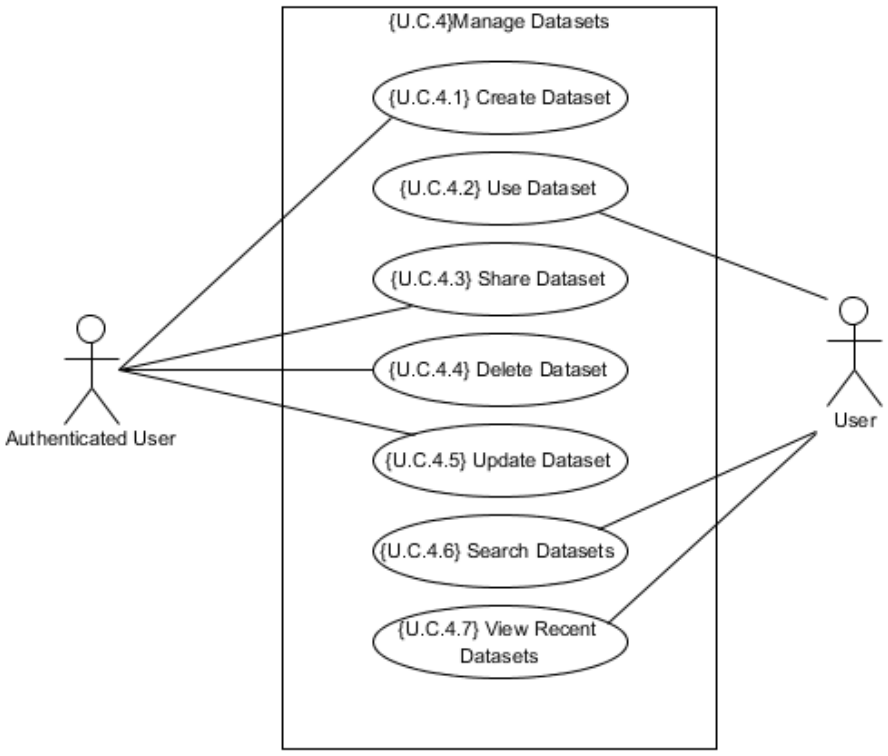


Figura 63 - *{U.C.4} Manage Datasets.*

Tabela 20 - Especificação do caso de uso *{U.C.4} Manage Datasets*.

{U.C.4} Manage Datasets.	
{U.C.4.1} Create Dataset	Permite criar *datasets:* • Permitir o *upload* de ficheiros *xls, xlsx, csv*; • Definir campos de atributos e de métricas. • Pré-visualizar *datasets.*
{U.C.4.2} Use Dataset	Permite usar os *datasets* e seus detalhes, criar gráficos e tabelas, relatório, criar *queries ad hoc* e agrupar por categorias; Após a seleção do *dataset*, torna-se possível usar as funcionalidades descritas em {U.C.3}, {U.C.4} e {U.C.5}.
{U.C.4.3} Share Dataset	Permite partilhar e retirar partilha de *datasets.* • Partilha com grupos de utilizadores; • Partilha com todos os utilizadores;
{U.C.4.4} Delete Dataset	Permite apagar *datasets*.

{U.C.4.5} Update Dataset	Permite alterar *datasets.*
{U.C.4.6} Search Datasets	Permite procurar *datasets* e agrupar por categorias.
{U.C.4.7} Recent Datasets	Permite visualizar *datasets* recentemente acedidos.

O caso de uso seguinte, *{U.C.5} Manage KPIs,* contém sete casos de uso folha como se pode verificar na Figura 64. A descrição deste caso de uso é apresentada abaixo na Tabela 21.

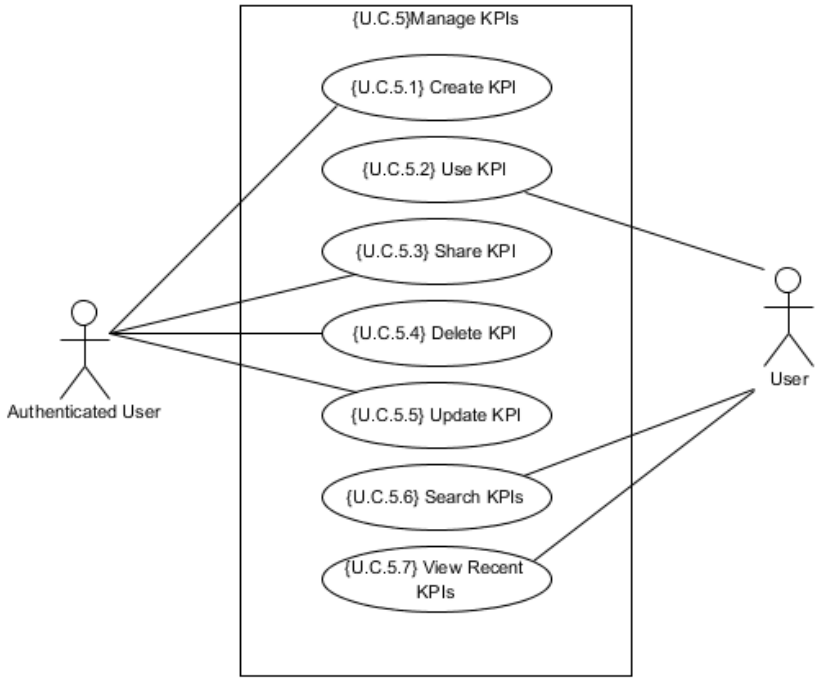


Figura 64 - *{U.C.5} Manage KPIs.*

Tabela 21 - Especificação do caso de uso *{U.C.5} Manage KPIs*.

{U.C.5} Manage KPIs.	
{U.C.5.1} Create KPI	Permite criar *kpi.*
{U.C.5.2} Use KPI	Permite usar *kpi* e agrupar por categorias.
{U.C.5.3} Share KPI	Permite partilhar e retirar partilha de *kpi.* • Partilha com grupos de utilizadores; • Partilha com todos os utilizadores;
{U.C.5.4} Delete KPI	Permite apagar *kpi*.
{U.C.5.5} Update KPI	Permite alterar *kpi.*
{U.C.5.6} Search KPIs	Permite procurar *kpis*.
{U.C.5.7} Recent KPIs	Permite visualizar *kpis* recentemente acedidos.